N72-13887

NASA TECHNICAL NOTE

NASA TN D-6534

NASA TN D-6534

AUTOMATED PRELIMINARY DESIGN OF SIMPLIFIED WING STRUCTURES TO SATISFY STRENGTH AND FLUTTER REQUIREMENTS

by W. Jefferson Stroud, Cornelia B. Dexter, and Manuel Stein

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • DECEMBER 1971

1. Report No.
NASA TN D-6534

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
AUTOMATED PRELIMINARY DESIGN OF SIMPLIFIED WING STRUCTURES TO SATISFY STRENGTH AND FLUTTER REQUIREMENTS

5. Report Date
December 1971

6. Performing Organization Code

7. Author(s)
W. Jefferson Stroud, Cornelia B. Dexter, and Manuel Stein

8. Performing Organization Report No.
L-7876

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23365

10. Work Unit No.
126-14-16-01

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A simple structural model of an aircraft wing is used to show the effects of strength (stress) and flutter requirements on the design of minimum-weight aircraft-wing structures. The wing is idealized as an isotropic sandwich plate with a variable cover thickness distribution and a variable depth between covers. Plate theory is used for the structural analysis, and piston theory is used for the unsteady aerodynamics in the flutter analysis. Mathematical programming techniques are used to find the minimum-weight cover thickness distribution which satisfies flutter, strength, and minimum-gage constraints. This report presents the method of solution, some sample results, and the computer program used to obtain these results.

The results indicate that the cover thickness distribution obtained when designing for the strength requirement alone may be quite different from the cover thickness distribution obtained when designing for either the flutter requirement alone or for both the strength and flutter requirements concurrently. This conclusion emphasizes the need for designing for both flutter and strength from the outset.

17. Key Words (Suggested by Author(s))
Optimization
Nonlinear programming
Mathematical programming
Strength and flutter design requirements
Wing design

18. Distribution Statement
Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	137	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

AUTOMATED PRELIMINARY DESIGN OF SIMPLIFIED WING STRUCTURES TO SATISFY STRENGTH AND FLUTTER REQUIREMENTS

By W. Jefferson Stroud, Cornelia B. Dexter,
and Manuel Stein
Langley Research Center

SUMMARY

A simple structural model of an aircraft wing is used to show the effects of strength (stress) and flutter requirements on the design of minimum-weight aircraft-wing structures. The wing is idealized as an isotropic sandwich plate with a variable cover thickness distribution and a variable depth between covers. Plate theory is used for the structural analysis, and piston theory is used for the unsteady aerodynamics in the flutter analysis. Mathematical programming techniques are used to find the minimum-weight cover thickness distribution which satisfies flutter, strength, and minimum-gage constraints. This report presents the method of solution, some sample results, and the computer program used to obtain these results.

The results indicate that the cover thickness distribution obtained when designing for the strength requirement alone may be quite different from the cover thickness distribution obtained when designing for either the flutter requirement alone or for both the strength and flutter requirements concurrently. This conclusion emphasizes the need for designing for both flutter and strength from the outset.

INTRODUCTION

The speed and storage capacity of modern-day computers, together with the development of efficient computer-oriented decision-making strategies, have encouraged researchers to seek ways to use the computer for not only analysis purposes but also for portions of the design process. Research is underway in this area because automated design methods have the potential for providing superior designs in a shorter time and at a lower cost than is possible with more conventional design techniques. One of the objectives of automated design is to incorporate interdisciplinary design requirements into the design process. The present report considers one example of an interdisciplinary design problem – that of designing a wing structure to meet a strength (stress level) requirement and a flutter requirement. In this example, an optimization procedure is used to determine the lightest structure which will meet these requirements.

The purpose of this report is to show the effects of strength (stress) and flutter requirements on the design of simplified aircraft-wing structures. The wing is idealized as an isotropic sandwich plate with a variable cover thickness distribution and a variable depth between covers. Plate theory is used for the structural analysis, and piston theory is used for the unsteady aerodynamics in the flutter analysis. Mathematical programming techniques are used to find the minimum-weight cover thickness distribution which satisfies flutter, strength, and minimum-gage constraints. This report presents the method of solution, some sample results, and the computer program used to obtain these results.

Some previous work dealing with the design of aircraft structures to satisfy aeroelastic constraints may be found in references 1 to 5. Some studies of automated structural design for strength, buckling, dynamic and/or deflection constraints are contained in references 6 to 12.

SYMBOLS

The physical quantities used in this paper are given in both the International System of Units (SI) (ref. 13) and in the U.S. Customary System of Units. The measurements and calculations were made in U.S. Customary Units. Appendix A presents factors relating these two systems of units.

a	speed of sound
C	design variables
c	local chord length
D	local flexural stiffness of wing
E	modulus of elasticity
h	total local depth of wing
M	Mach number
m	mass of wing per unit area
N	number of spanwise stations on the wing (see fig. 3)
P	sum of the objective function and the penalty function (see eq. (16))
p	normal loading per unit area

r	penalty-function weighting factor (see eq. (16))
t	local thickness of wing covers
v	velocity
v_f	flutter velocity
W,w	assumed normal deflection of wing neutral surface (see eqs. (1) and (2))
X,Y,Z	chordwise, spanwise, and normal axes, respectively
x,y	chordwise and spanwise coordinates, respectively
x_L	value of x at wing leading edge
x_T	value of x at wing trailing edge (zero for all calculations presented herein)
y_S	value of y at wing tip, semispan
α	real part of flutter determinant
β	imaginary part of flutter determinant
γ	ratio of specific heats
ϵ	spanwise finite-difference increment
$\epsilon_x,\epsilon_y,\gamma_{xy}$	components of normal and shearing strain in wing cover plates
μ	Poisson's ratio
ρ	mass density of air
σ_{cr}	critical value of the stress defined by the Von Mises stress condition
$\sigma_x,\sigma_y,\tau_{xy}$	components of normal and shearing stress in wing cover plates
τ	time

φ_i deflection coefficients (see eq. (2))

ω frequency, rad/s

ω_f flutter frequency

Dots indicate differentiation with respect to time.

ANALYSIS

The analysis procedures used in this report are plate theory for the wing structural analysis, piston theory for the unsteady aerodynamics in the flutter analysis, and mathematical programming techniques for finding the minimum-weight design. These three procedures are discussed briefly in the next three sections. Additional discussion and detailed definitions of matrix expressions are given in appendix B.

Plate Theory for Wing Structural Analysis

The wing is assumed to be sufficiently thin to be analyzed with plate theory. It is further assumed that internal bulkheads, ribs, and spars exist which prevent transverse shear deflections, but any other load-carrying capacity of these structural elements is neglected. All load is assumed to be carried by a set of wing cover plates – an upper cover plate and a lower cover plate. The wing structural analysis is concerned only with these cover plates. The wing is assumed to have a symmetric biconvex airfoil shape and, therefore, the distance between covers varies over the surface of the wing (fig. 1). The covers have a variable thickness distribution which is adjusted in the design procedure to provide the minimum-weight design.

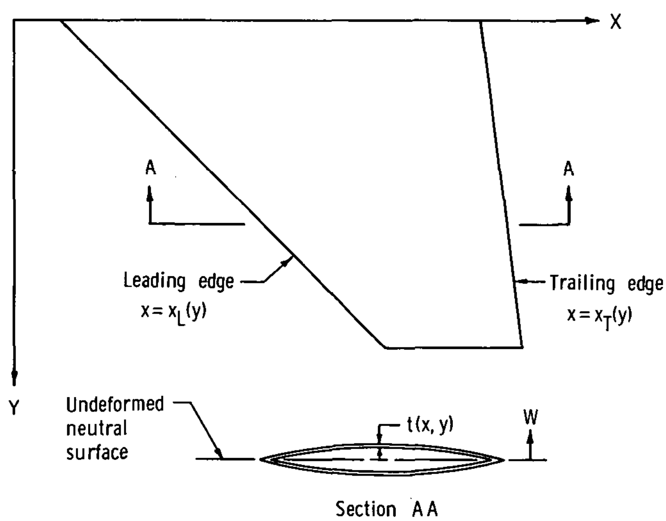


Figure 1.- Wing coordinate system.

The wing structural analysis is based upon the approach presented in reference 14. The normal deflection W of the wing neutral surface is assumed to be given by

$$W(x,y,\tau) = w(x,y)e^{i\omega\tau} \tag{1}$$

where

$$w(x,y) = \varphi_0(y) + x\varphi_1(y) + x^2\varphi_2(y) \tag{2}$$

The assumed solution includes terms only to the second degree in the x (chordwise) direction. Finite differences in the y (spanwise) direction are used to solve for the unknowns $\varphi_0(y)$, $\varphi_1(y)$, and $\varphi_2(y)$. The wing coordinate system and deflection shape are shown in figures 1 and 2, respectively. The location of the Y-axis is arbitrary.

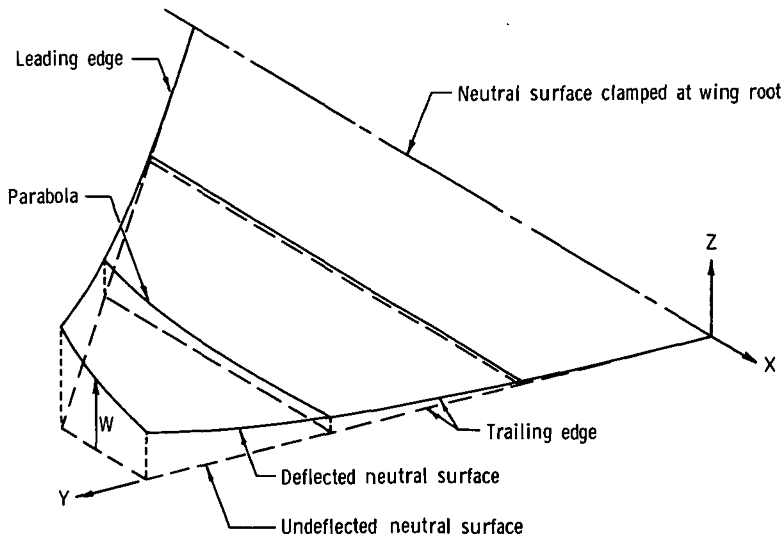


Figure 2.- Diagram of undeflected and deflected neutral surface of wing showing assumed parabolic chordwise-deflection shape.

An energy approach (ref. 14) is used to obtain the equilibrium equations for a variable-thickness isotropic plate or for an isotropic built-up plate with variable depth. The matrix form of these equilibrium equations is

$$\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{bmatrix} = \epsilon^4 \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} \tag{3}$$

in which the matrix $\left[A_{ij}\right]$ is composed of terms involving the stiffness and derivatives of the stiffness, the vector $\left[\varphi_i\right]$ describes the deflections, the vector $\left[p_i\right]$ is the normal loading, and ϵ is the spanwise finite difference increment shown in figure 3. The matrix expressions are presented in detail in appendix B. The normal loading used in the strength analysis is a uniform normal loading. The wing is assumed to be fixed at the root.

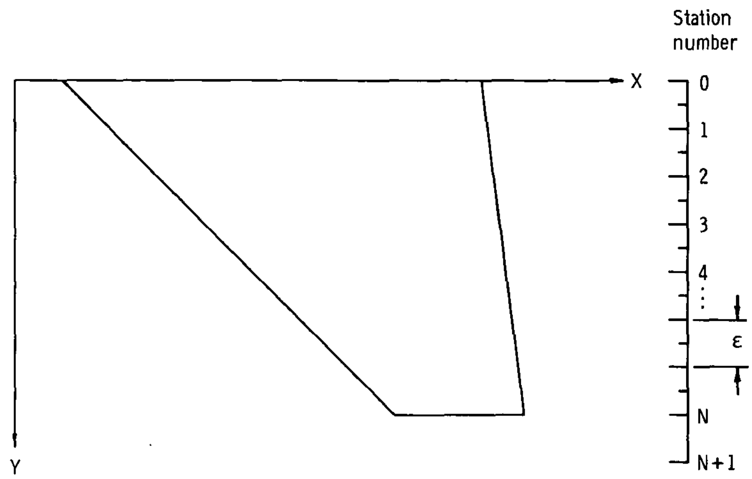


Figure 3.- Location of stations on planform of half-wing.

Once the deflections have been calculated from equations (3) and (2), the strains in the wing covers are calculated by using the following strain-displacement relations:

$$\epsilon_x = \mp \frac{h}{2} \frac{\partial^2 w}{\partial x^2} \tag{4}$$

$$\epsilon_y = \mp \frac{h}{2} \frac{\partial^2 w}{\partial y^2} \tag{5}$$

$$\gamma_{xy} = \mp h \frac{\partial^2 w}{\partial x \, \partial y} \tag{6}$$

where the upper and lower signs are for the upper and lower wing covers, respectively, and h is the local distance between the upper wing-cover midplane and the lower wing-cover midplane. The stresses can then be calculated by using the following stress-strain relations:

$$\sigma_x = \frac{E}{1 - \mu^2}\left(\epsilon_x + \mu\epsilon_y\right) \tag{7}$$

$$\sigma_y = \frac{E}{1 - \mu^2}\left(\epsilon_y + \mu\epsilon_x\right) \tag{8}$$

$$\tau_{xy} = \frac{E}{2(1 + \mu)} \gamma_{xy} \tag{9}$$

These stresses are used in the strength constraint discussed in a subsequent section entitled "Mathematical Programming Techniques for Determining Minimum-Weight Design."

Piston Theory for Unsteady Aerodynamics in Flutter Analysis

In contrast to the uniform normal loading used in the strength analysis of the wing, the normal loading used in the flutter analysis is a function of x, y, and τ (time). The total loading that acts on an airfoil in motion is the sum of the inertia and unsteady aerodynamic loadings. (Structural damping is neglected.) Based on second-order piston theory for symmetrical airfoils (refs. 15 and 16), the sum of these two loadings is

$$p(x,y,\tau) = -m\ddot{W} - 2\rho a\left(1 + \frac{\gamma + 1}{4} M \frac{\partial h}{\partial x}\right)\left(\frac{\partial}{\partial \tau} + v \frac{\partial}{\partial x}\right)W \tag{10}$$

The unsteady loading $p(x,y,\tau)$ can be taken to the left side of the equilibrium equation (eq. (3)), thereby allowing the equation to be written in the form

$$\left\{\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} + \epsilon^4 \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \\ B_{20} & B_{21} & B_{22} \end{bmatrix}\right\} \begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{bmatrix} = 0 \tag{11}$$

Detailed expressions which define the components of equation (11) are given in appendix B.

The flutter speed v_f and the flutter frequency ω_f are calculated by setting the complex determinant of the matrix of coefficients of φ_i in equation (11) equal to zero — that is,

$$\left|\left[A_{ij} + \epsilon^4 B_{ij}\right]\right| = \alpha + i\beta = 0 \tag{12}$$

There are, therefore, two equations

$$\alpha = 0 \tag{13}$$

$$\beta = 0 \tag{14}$$

in the two unknowns v_f and ω_f. First, the solution neighborhood for equation (12) is established by evaluating the determinant for many combinations of ω and v. The Newton-Raphson method is then used to determine ω_f and v_f precisely. An example of this solution technique is given in appendix C.

Mathematical Programming Techniques for Determining Minimum-Weight Design

The thickness distribution of the wing covers is adjusted to give the minimum-weight design which satisfies all the design requirements. The thickness distribution is assumed to be of the form

$$t(x,y) = \sum_i \sum_j C_{ij} x^i y^j \tag{15}$$

The coefficients C_{ij} are design variables which are adjusted to give the minimum-weight design. Henceforth, the design variables will be referred to as the vector of design variables denoted by $\vec{C}$. The detailed expressions used for $t(x,y)$ are given in a subsequent section entitled "Applications and Results."

In this report the constraints, which are the design requirements, are introduced into the problem by way of an interior penalty function. This method of treating constraints is sometimes called "SUMT" – Sequential Unconstrained Minimization Technique (refs. 17 and 18). In SUMT, the objective function, which is taken to be the weight in this study, is augmented by a penalty term that accounts for the constraints. The combination of these two terms is called the P function:

$$P\left(\vec{C},r\right) = \text{Weight}\left(\vec{C}\right) + r \sum_j \frac{1}{g_j\left(\vec{C}\right)} \tag{16}$$

The term $r \sum_j \frac{1}{g_j\left(\vec{C}\right)}$ is the penalty term in which r is a weighting factor and $g_j\left(\vec{C}\right)$

are the inequality constraints. The inequality constraints are formulated in such a way that they may be written as

$$g_j(\vec{C}) > 0 \tag{17}$$

The penalty term repels the P function from the inequality constraints and transforms the problem from a constrained minimization problem to an unconstrained problem. By carrying out a sequence of minimizations of the P function corresponding to a sequence of reductions in the value of the weighting factor r, it is possible to obtain the vector of design variables which satisfies the constraints and gives the minimum value of the objective function. The constraints used herein are as follows:

(1) For flutter, the flutter speed v_f shall not be less than some prescribed critical flutter speed $v_{f,cr}$. This constraint can be expressed mathematically as

$$g_1 \equiv 1 - \frac{v_{f,cr}}{v_f} > 0 \tag{18}$$

(2) For strength, at no point on the wing covers shall the stress exceed the prescribed critical stress, or

$$g_2 \equiv \frac{1}{\frac{1}{A}\iint_A \frac{dA}{1 - \frac{\sigma}{\sigma_{cr}}}} > 0 \tag{19}$$

where A is the wing area,

$$\sigma = \left(\sigma_x^2 - \sigma_x\sigma_y + \sigma_y^2 + 3\tau_{xy}^2\right)^{1/2}$$

which is the Von Mises stress condition, and σ_{cr} is the critical value of σ.

(3) For minimum gage, at no point on the wing covers shall the thickness t be less than some prescribed minimum thickness t_{min}, or

$$g_3 \equiv \frac{1}{\frac{1}{A}\iint_A \frac{dA}{1 - \frac{t_{min}}{t}}} > 0 \tag{20}$$

Once a set of design variables has been defined (eq. (15)), and the design requirements have been formulated as constraints (eqs. (18) to (20)), the wing can be designed by using mathematical programming methods. Additional comments on the math programming approach used are given in appendix B.

A computer program denoted "SWIFT" which incorporates the structural analysis, flutter analysis, and optimization techniques just discussed is presented in appendix C.

In order to check the validity of the structural analysis, natural-vibration frequencies for a trapezoidal flat plate were calculated by using the preceding analysis and are compared with experimentally determined natural frequencies in appendix D. A theoretical flutter speed and frequency for that plate are also presented in appendix D but are not compared with experimental results.

APPLICATIONS AND RESULTS

In order to show the effects of flutter and strength requirements on the design of aircraft wings, these requirements were used in the calculation of several minimum-weight wing cover thickness distributions. In these sample calculations, the number and choice of design variables is varied and the design requirements are varied. The wing properties used in all calculations are presented in figure 4. For the examples presented herein, the Y-axis is chosen to lie along the wing trailing edge. The number of spanwise stations N used in the structural analysis is six.

The following specific examples were considered:

First, the following set of design requirements was chosen:

(1) For flutter, the wing is to be flutter-free at a dynamic pressure of 160 kN/m^2 (3340 lbf/ft^2) at an altitude of 7620 meters (25 000 ft). (This corresponds to $M = 2.46$.)

(2) For strength, the wing is to support a uniformly distributed loading of 6.89 kN/m^2 (144 lbf/ft^2).

(3) The minimum-gage requirement is 0.0508 centimeter (0.0200 in.).

These design requirements, together with a sixteen-design-variable polynomial, were used to obtain a flutter design (requirements 1 and 3), a strength design (requirements 2 and 3), and a design (denoted as a "strength-flutter design" or "combined design") which meets the flutter, strength, and minimum-gage requirements.

Second, these same design requirements, together with a six-design-variable polynomial, were also used to obtain a flutter design, a strength design, and a strength-flutter design.

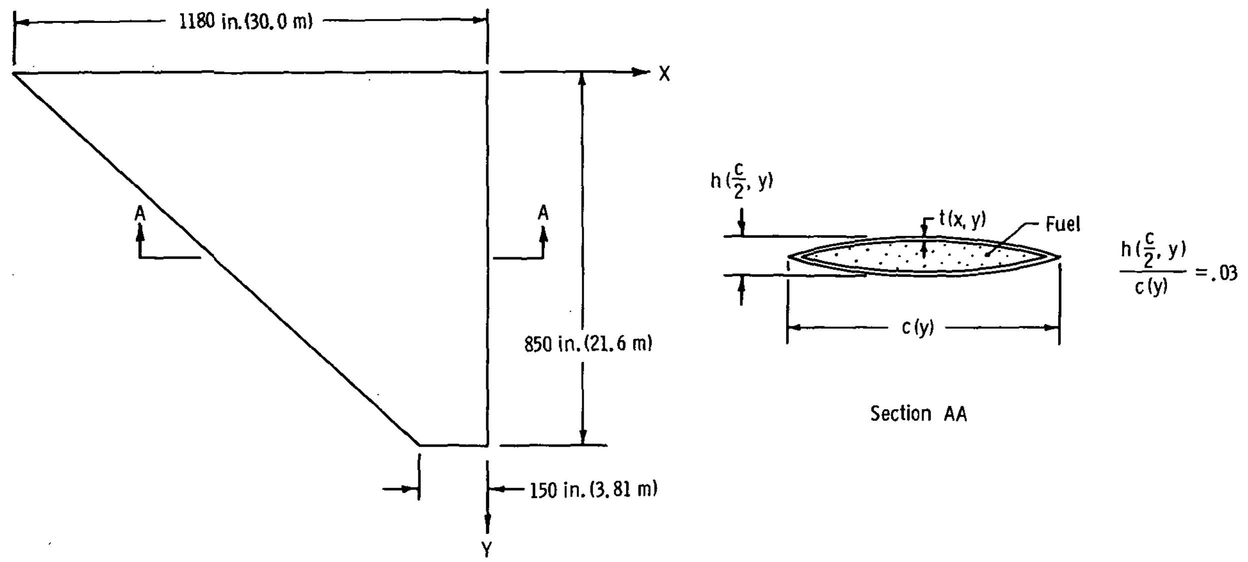


Figure 4.- Wing description and cover-material properties used in the sample calculations. Fuel mass, 150 000 lbm (68.0 Mg). Properties of titanium material: $E = 16\,400$ ksi (113 GN/m^2); $\mu = 0.3$; $\sigma_{cr} = 125$ ksi (863 MN/m^2); density, 0.160 lbm/in^3 (4430 kg/m^3).

Third, the flutter and strength design requirements were varied and the mass of minimum-weight wings designed by using the six-design-variable polynomial was calculated as a function of the design requirements.

Results Obtained With Sixteen Design Variables

The thickness distribution chosen for the first example is a sixteen-term polynomial. That polynomial is

$$\begin{aligned} t = {} & C_1 + C_2\left(1 - \frac{y}{y_S}\right) + C_3\left(1 - \frac{y}{y_S}\right)^2 + C_4\left(1 - \frac{y}{y_S}\right)^3 \\ & + \left[C_5 + C_6\left(1 - \frac{y}{y_S}\right) + C_7\left(1 - \frac{y}{y_S}\right)^2 + C_8\left(1 - \frac{y}{y_S}\right)^3\right]\left(1 - \frac{2x}{x_L}\right) \\ & + \left[C_9 + C_{10}\left(1 - \frac{y}{y_S}\right) + C_{11}\left(1 - \frac{y}{y_S}\right)^2 + C_{12}\left(1 - \frac{y}{y_S}\right)^3\right]\left(1 - \frac{2x}{x_L}\right)^2 \\ & + \left[C_{13} + C_{14}\left(1 - \frac{y}{y_S}\right) + C_{15}\left(1 - \frac{y}{y_S}\right)^2 + C_{16}\left(1 - \frac{y}{y_S}\right)^3\right]\left(1 - \frac{2x}{x_L}\right)^3 \end{aligned} \tag{21}$$

in which x is the chordwise coordinate, y is the spanwise coordinate, x_L is the value of x at the wing leading edge, y_S is the semispan, and C_i $(i = 1,2,3,. . .,16)$ are the design variables. For clarity, the spanwise and chordwise variations corresponding to each term in equation (21) are shown in figure 5. The thickness distribution for the upper cover is the same as for the lower cover.

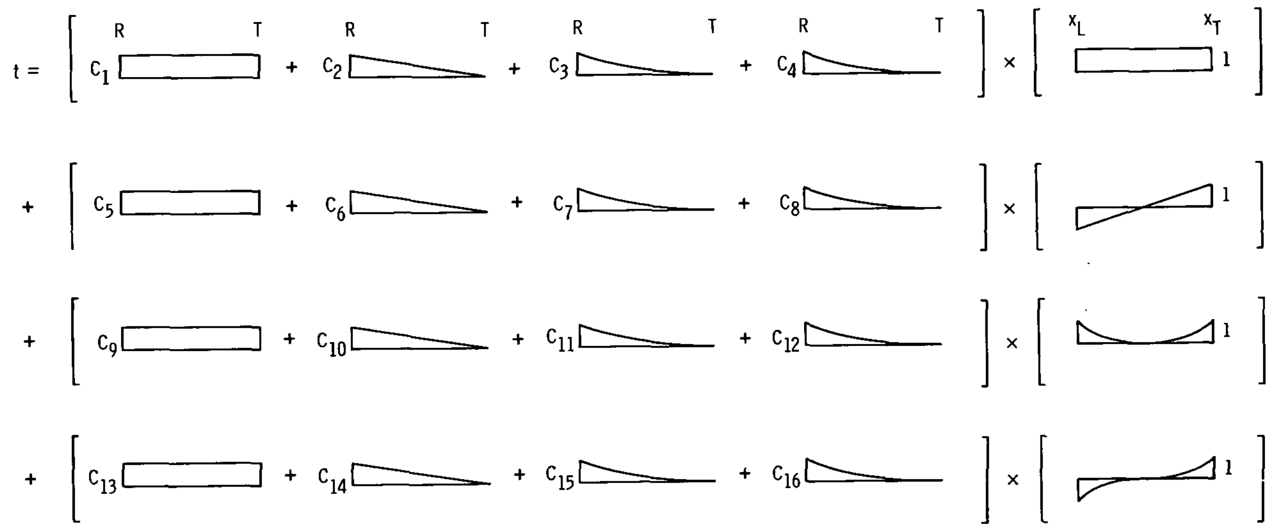


Figure 5.- Spanwise and chordwise variations for each term in equation (21). R = root; T = tip; x_L = leading edge; x_T = trailing edge.

Contour plots showing wing cover thickness distributions for the flutter, strength, and combined designs are shown in figure 6. These plots show contours for a single cover. If only the flutter and minimum-gage requirements are considered, the lightest cover thickness distribution that can be obtained from the sixteen design variables described in equation (21) is shown in figure 6(a). The mass of the semispan wing (that is, the mass of both the top and the bottom cover) is 3160 kilograms (6960 lbm). This flutter design will support a uniform static pressure loading of only 2.82 kN/m^2 (58.9 lbf/ft^2). If only the strength and minimum-gage requirements are considered, the lightest cover thickness distribution for the sixteen prescribed design variables is shown in figure 6(b). The cover mass is 3220 kilograms (7100 lbm). This strength design will flutter at a dynamic pressure of 51.2 kN/m^2 (1070 lbf/ft^2). The cover thickness distribution for the flutter design is quite different from that for the strength design. The design which satisfies both the flutter and strength requirement is shown in figure 6(c). The cover mass for this design is 4030 kilograms (8890 lbm).

Because of the large mass of fuel (68.0 megagrams (150 000 lbm)) carried in the wing, the contribution of the wing covers to the total mass distribution is small but is, nevertheless, included in the flutter analysis used to obtain these results. The primary effect of the thickness distribution is to establish the stiffness distribution.

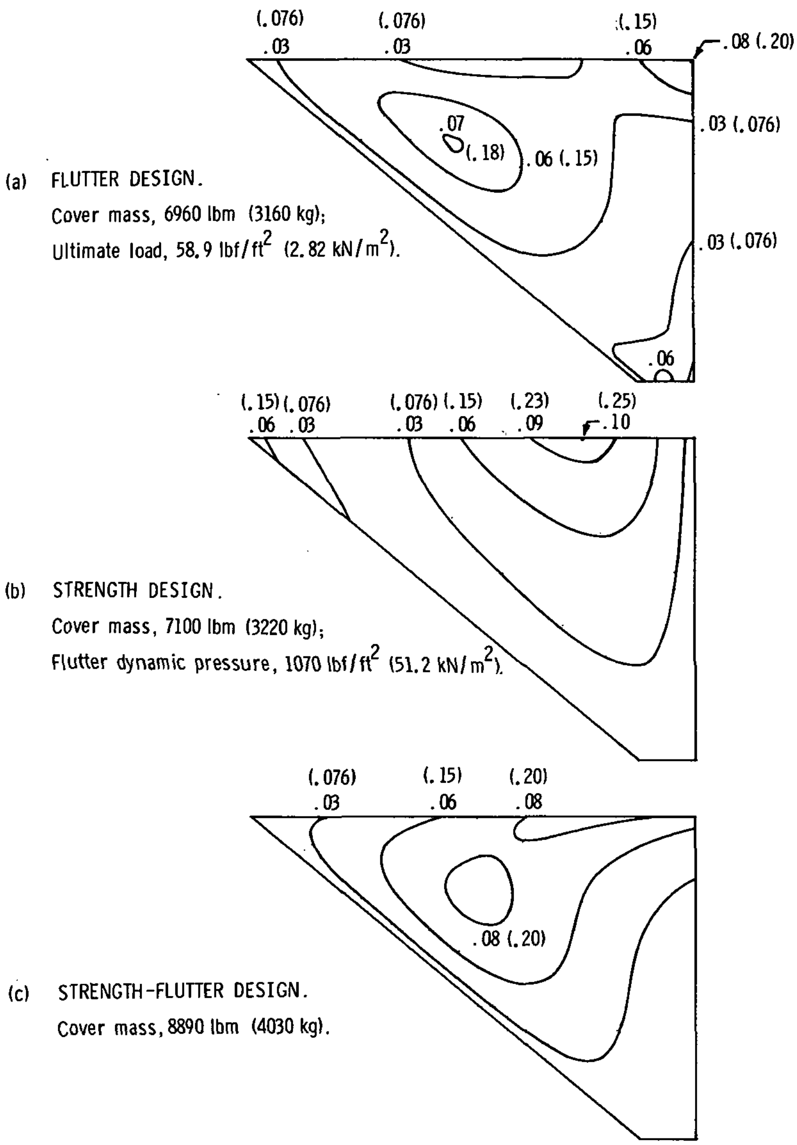


Figure 6.- Contour plots showing wing cover thickness distributions for flutter, strength, and combined designs obtained by using sixteen design variables. Contour labels are in inches (centimeters).

Results Obtained With Six Design Variables

Calculations were also made with a judicious combination of certain terms in equation (21). The thickness expression which was used is

$$t = t_{min} + \left[C_1\left(1 - \frac{y}{y_S}\right) + C_2\left(1 - \frac{y}{y_S}\right)^2 + C_3\left(1 - \frac{y}{y_S}\right)^3\right]\left[1 - \left(1 - \frac{2x}{x_L}\right)^2\right]$$
$$+ \left[C_4\left(1 - \frac{y}{y_S}\right) + C_5\left(1 - \frac{y}{y_S}\right)^2 + C_6\left(1 - \frac{y}{y_S}\right)^3\right]\left[1 - \left(1 - \frac{2x}{x_L}\right)^2\right]\left(1 - \frac{2x}{x_L}\right) \tag{22}$$

in which t_{min} = 0.0508 centimeter (0.0200 in.), the minimum-gage requirement. The spanwise and chordwise variations for each term in equation (22) are shown in figure 7. The thickness distribution described by equation (22) is minimum gage along the leading and trailing edges and at the tip – a reasonable limitation.

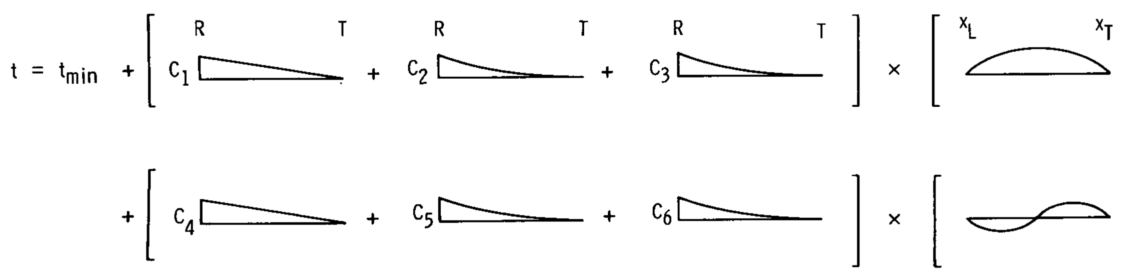
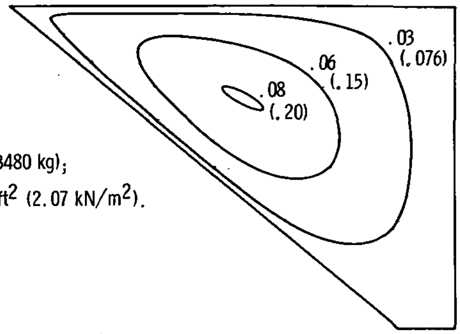


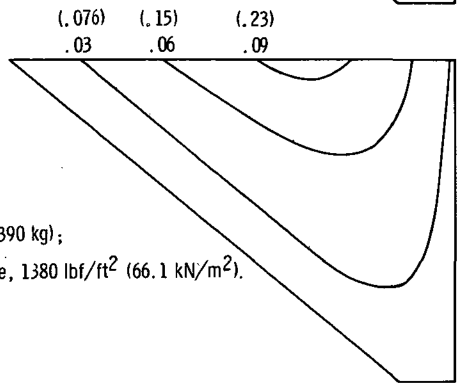
Figure 7.- Spanwise and chordwise variations for each term in equation (22). R = root; T = tip; x_L = leading edge; x_T = trailing edge.

All of the thickness variations described by equation (22) can also be obtained from equation (21). Therefore, the design obtained by using equation (22) cannot be lighter than the design obtained by using equation (21). Equation (22), however, contains only six design variables, compared with sixteen design variables in equation (21). Since computer time for the design process depends strongly upon the number of design variables, it is advantageous to use a small number of well-chosen design variables rather than a large number of arbitrarily chosen design variables.

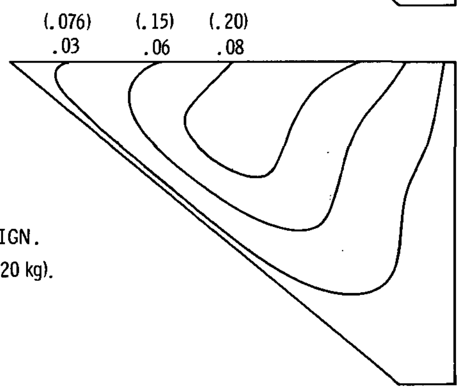
The design requirements used for this second example are the same as those previously stated. Contour plots showing wing cover thickness distributions for the flutter, strength, and combined designs are shown in figures 8(a), 8(b), and 8(c), respectively. As in the preceding example, the cover thickness distribution for the flutter design is quite different from that for the strength design.



(a) FLUTTER DESIGN.
Cover mass, 7670 lbm (3480 kg);
Ultimate load, 43.2 lbf/ft^2 (2.07 kN/m^2).



(b) STRENGTH DESIGN.
Cover mass, 7470 lbm (3390 kg);
Flutter dynamic pressure, 1380 lbf/ft^2 (66.1 kN/m^2).



(c) STRENGTH-FLUTTER DESIGN.
Cover mass, 9090 lbm (4120 kg).

Figure 8.- Contour plots showing wing cover thickness distributions for flutter, strength, and combined designs obtained by using six design variables. Contour labels are in inches (centimeters).

For the flutter design, the new six-design-variable thickness distribution is 10 percent heavier than the sixteen-design-variable thickness distribution. The new strength design is only 5 percent heavier than the sixteen-design-variable case. The new combined design is only 2 percent heavier than the sixteen-design-variable case.

Results Obtained for Various Strength and Flutter Design Requirements

The wing properties given in figure 4 were used in calculating several other minimum-weight thickness distributions for various strength and flutter design requirements. All of these calculations were made with the six-design-variable polynomial

given in equation (22). The results are presented in figure 9 in which minimum-weight wing cover mass is shown as a function of flutter (abscissa) and strength (ordinate) design requirements. Lines of constant cover mass are shown as dashed lines. The solid curve which is almost horizontal represents designs dictated by flutter requirements, and the solid curve that is almost vertical represents designs dictated by strength requirements. Points a, b, and c indicate the flutter, strength, and combined designs shown in figures 8(a), 8(b), and 8(c), respectively.

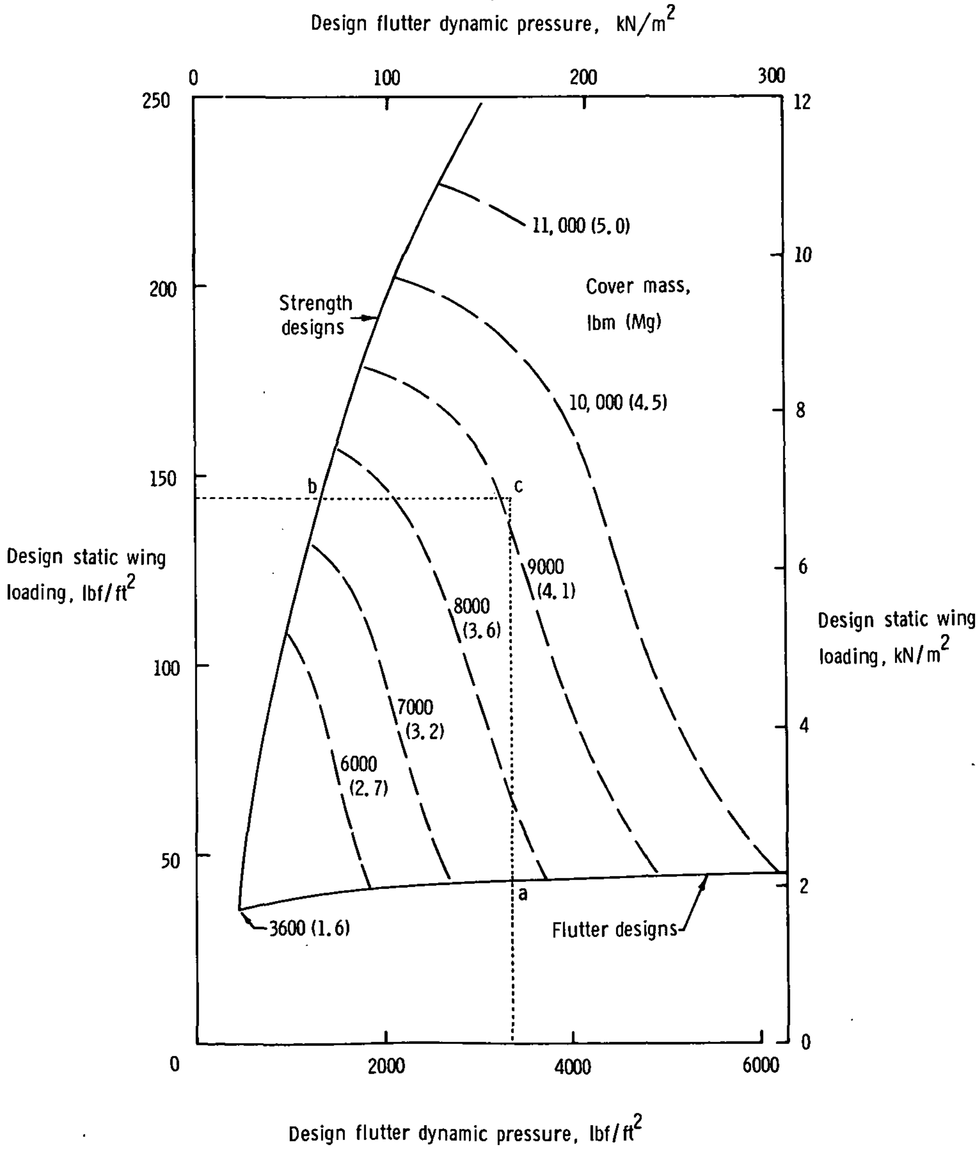


Figure 9.- Minimum-weight wing cover mass as a function of design flutter dynamic pressure (flutter requirement) and design static wing loading (strength requirement).

The results presented in figure 9 indicate that for this example there is little interaction between strength and flutter design requirements. For a strength-designed wing, improving the strength in the best possible way improves the flutter characteristics only slightly. For a flutter-designed wing, increasing the flutter speed in the best possible way has a negligible effect on the strength. The fact that designing for strength is quite different from designing for flutter emphasizes the need for designing for both strength and flutter from the outset.

There are many ways to obtain a design which satisfies both the flutter and strength design requirements given by point c in figure 9. One approach is to start with the strength design shown at point b and increase the thickness uniformly over the entire wing until the flutter requirement is satisfied. For the present set of design conditions, the resulting design is 38 percent heavier than the minimum-weight combined design at point c.

A second approach is to use the envelope of the thickness distributions for the flutter and strength cases. That is, at each point on the wing the cover thickness is taken to be the larger of the two following thicknesses: (1) the thickness at that point for the flutter design, and (2) the thickness at that point for the strength design. In the present case, the resulting envelope design does meet the design requirements and is only about 3 percent heavier than the combined design at point c. However, all envelope designs do not meet the design requirements. For example, if the strength design requirement given by point c in figure 9 is reduced 50 percent to 3.45 kN/m^2 (72 lbf/ft^2) without changing the flutter design requirement, the envelope design corresponding to the new set of design requirements is about 2 percent lighter than the minimum-weight combined design for that same set of design requirements. However, the envelope design flutters at a dynamic pressure about 9 percent lower than the flutter design requirement.

A third possible approach is to use the strength design as a minimum-gage requirement and, by using an optimization technique, add material to meet the flutter requirement. Such a design would be lighter than an acceptable envelope design. There are, of course, many other approaches.

The intent of this report is not to prove that a combined design approach as characterized by the math programming techniques used herein is superior to any other design approach. There are, however, two items to consider with respect to the combined design approach. First, for the set of design variables considered, the combined design approach will give the lightest design that meets all design requirements. In this simple model which combines two disciplines – structural analysis and aeroelasticity – the weight penalty caused by designing separately and sequentially rather than concurrently may be small. However, as the model becomes more complex and other disciplines are added, that penalty may increase. Second, for the optimization techniques described herein it is

faster to carry out a combined design than to design for strength and then to design for flutter. It is realized, however, that there are specialized design techniques that are tailored for a particular discipline. These techniques may be well-suited for sequential design and may provide an adequate design with shorter computer run times. The effectiveness of these design techniques can be assessed by comparing the results obtained by using these techniques with the results obtained by using a combined-design math-programming approach.

CONCLUDING REMARKS

A simple structural model of an aircraft wing has been used to show the effects of strength and flutter requirements on the design of minimum-weight wing structures. The wing structural analysis is based on plate theory. Piston theory is used in the flutter analysis. This report has presented the method of solution, some sample results, and the computer program used to obtain these results.

Wings were designed by adjusting the coefficients of polynomials which describe the wing cover thickness distribution. Sample calculations were carried out for a polynomial containing six coefficients (design variables) and for a polynomial containing sixteen coefficients. Mathematical programming techniques were used to determine the values of the coefficients which gave the lightest cover thickness distribution that satisfied flutter, strength, and minimum-gage requirements.

The results indicate that the cover thickness distribution obtained when designing for the strength requirement alone may be quite different from the cover thickness distribution obtained when designing for either the flutter requirement alone or for both the strength and flutter requirements concurrently. This conclusion emphasizes the need for designing for both flutter and strength from the outset.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., October 27, 1971.

APPENDIX A

CONVERSION OF U.S. CUSTOMARY UNITS TO SI UNITS

The International System of Units (SI) was adopted by the Eleventh General Conference on Weights and Measures, Paris, October 1960 (ref. 13). Conversion factors for the units used herein are given in the following table:

Physical quantity	U.S. Customary Unit	Conversion factor (*)	SI Unit (**)
Force	lbf	4.44822	newtons (N)
Length	ft	0.3048	meters (m)
	in.	0.0254	
Mass	lbm	0.45359	kilograms (kg)
Stress, modulus . . .	ksi	6.895×10^6	newtons/meter^2 (N/m^2)
Density	lbm/in^3	27.68×10^6	grams/meter^3 (g/m^3)
Pressure	lbf/ft^2	47.88	newtons/meter^2 (N/m^2)
Speed	ft/s	0.3048	meters/second (m/s)

*Multiply value given in U.S. Customary Units by conversion factor to obtain equivalent value in SI Units.

**Prefixes to indicate multiples of units are as follows:

Prefix	Multiple
giga (G)	10^9
mega (M)	10^6
kilo (k)	10^3
deci (d)	10^{-1}
centi (c)	10^{-2}
milli (m)	10^{-3}

APPENDIX B

DEFINITIONS OF MATRIX EXPRESSIONS AND FURTHER DISCUSSION OF ANALYSIS

Plate Theory for Wing Structural Analysis

For convenience, the plate equilibrium equation (eq. (3)) is repeated here

$$\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{bmatrix} = \epsilon^4 \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} \tag{B1}$$

The matrix components of equation (B1) are

$$\left.\begin{aligned}
\left[A_{00}\right] &= \left[D_0\right]\left[a_0\right]\left[D_0\right]' \\
\left[A_{01}\right] &= \left[D_0\right]\left[a_1\right]\left[D_0\right]' \\
\left[A_{02}\right] &= \left[D_0\right]\left(\left[a_2\right]\left[D_0\right]' + 2\mu\epsilon^2\left[\bar{a}_0\right]\right) \\
\left[A_{11}\right] &= \left[D_0\right]\left[a_2\right]\left[D_0\right]' + 2(1-\mu)\epsilon^2\left[D_3\right]\left[a_0^*\right]\left[D_3\right]' \\
\left[A_{12}\right] &= \left[D_0\right]\left[a_3\right]\left[D_0\right]' + 4(1-\mu)\epsilon^2\left[D_3\right]\left[a_1^*\right]\left[D_3\right]' + 2\mu\epsilon^2\left[D_0\right]\left[\bar{a}_1\right] \\
\left[A_{22}\right] &= \left[D_0\right]\left[a_4\right]\left[D_0\right]' + 8(1-\mu)\epsilon^2\left[D_3\right]\left[a_2^*\right]\left[D_3\right]' + 2\mu\epsilon^2\left\{\left[D_0\right]\left[\bar{a}_2\right]\right. \\
&\qquad \left. + \left(\left[D_0\right]\left[\bar{a}_2\right]\right)'\right\} + 4\epsilon^4\left[\hat{a}_0\right] \\
\left[A_{10}\right] &= \left[A_{01}\right]' \qquad \left[A_{20}\right] = \left[A_{02}\right]' \qquad \left[A_{21}\right] = \left[A_{12}\right]'
\end{aligned}\right\} \tag{B2}$$

Primes indicate the transpose of a matrix, and ϵ is the finite-difference increment in the spanwise direction. The finite-difference increments are shown in figure 3.

In this report the wing is assumed to be fixed at the root. Because more general boundary conditions are considered in reference 14, some of the expressions used herein differ slightly from the comparable expressions in reference 14. For the most part, however, the quantities defined herein are the same as those defined in reference 14. The matrices in equations (B2) are defined as follows (where the number of rows and columns, in that order, are indicated below the equation numbers):

$$\left[D_0\right] = \begin{bmatrix} 2 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \\ & & & & & & 1 \end{bmatrix} \qquad \begin{matrix} \text{(B3)} \\ (N+1)\times(N+1) \end{matrix}$$

$$\left[D_3\right] = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \cdot & \cdot & \cdot \\ & & & & 1 & -1 \\ & & & & & 1 \\ & & & & & 0 \end{bmatrix} \qquad \begin{matrix} \text{(B4)} \\ (N+1)\times N \end{matrix}$$

$$\underset{k=0,1,2,3,4}{\left[a_k\right]} = \begin{bmatrix} \frac{1}{2}a_{k,0} & & & & & \\ & a_{k,1} & & & & \\ & & a_{k,2} & & & \\ & & & \cdots & & \\ & & & & a_{k,N-1} & \\ & & & & & \frac{1}{2}a_{k,N} \end{bmatrix} \qquad \begin{matrix} \text{(B5)} \\ (N+1)\times(N+1) \end{matrix}$$

$$\underset{k=0,1,2}{\left[\bar{a}_k\right]} = \begin{bmatrix} 0 & & & & & & \\ a_{k,1} & & & & & & \\ & a_{k,2} & & & & & \\ & & \cdots & & & & \\ & & & a_{k,N-1} & & & \\ & & & & \frac{1}{2}\,a_{k,N} & 0 \end{bmatrix} \tag{B6}$$

$(N+1)\times(N+1)$

$$\underset{k=0,1,2}{\left[a_k^*\right]} = \begin{bmatrix} a_{k,\frac{1}{2}} & & & \\ & a_{k,\frac{3}{2}} & & \\ & & \cdots & \\ & & & a_{k,N-\frac{1}{2}} \end{bmatrix} \tag{B7}$$

$N\times N$

$$\left[\hat{a}_0\right] = \begin{bmatrix} a_{0,1} & & & & & \\ & a_{0,2} & & & & \\ & & \cdots & & & \\ & & & a_{0,N-1} & & \\ & & & & \frac{1}{2}\,a_{0,N} & \\ & & & & & 0 \end{bmatrix} \tag{B8}$$

$(N+1)\times(N+1)$

in which

$$a_k(y) = \int_{x_L(y)}^{x_T(y)} Dx^k\,dx \tag{B9}$$

$$D = \frac{Eth^2}{2\left(1 - \mu^2\right)} + \frac{2Et^3}{12\left(1 - \mu^2\right)} \tag{B10}$$

and

E modulus of elasticity

t local thickness of either the upper or lower cover (airfoils are symmetrical)

h total local depth of wing

μ Poisson's ratio

The quantity $a_{k,0}$ is the value of a_k at the root, $y = 0$; $a_{k,\frac{1}{2}}$ is a_k at $y = \frac{1}{2} \cdot \epsilon$; $a_{k,1}$ is a_k at $y = 1 \cdot \epsilon$; and $a_{k,N}$ is a_k at the tip. The wing tip is located at $y = N \cdot \epsilon$. (See fig. 3.) Several of the quantities which follow are defined in the same way.

The column vectors in equation (B1) describe the displacements and loading and are given by

$$\underset{k=0,1,2}{\begin{bmatrix}\varphi_k\end{bmatrix}} = \begin{bmatrix}\varphi_{k,1}\\ \varphi_{k,2}\\ \cdot\\ \cdot\\ \cdot\\ \varphi_{k,N+1}\end{bmatrix} \tag{B11}$$

and

$$\underset{k=0,1,2}{\begin{bmatrix}p_k\end{bmatrix}} = \begin{bmatrix}p_{k,1}\\ p_{k,2}\\ \cdot\\ \cdot\\ \cdot\\ p_{k,N}\\ 0\end{bmatrix} \tag{B12}$$

in which

$$p_k(y) = \int_{x_L(y)}^{x_T(y)} px^k \, dx \tag{B13}$$

This definition of p_k differs from that given in reference 11. As a result, there is a factor ϵ^4 in equation (B1) compared with a factor ϵ^3 in the equivalent equation (eq. (30)) of reference 14. The quantity p in equation (B13) is the intensity of the normal loading and has units of force per unit area.

In the present report the static normal loading used in the strength analysis of the wing is assumed to be a uniform loading; therefore,

$$p_k(y) = p\int_{x_L(y)}^{x_T(y)} x^k \, dx \tag{B14}$$

This completes the definitions of terms that appear in equation (B1). The deflections are calculated from equations (B1) and (2), and the strains are calculated from derivatives of the deflections (eqs. (4) to (6)). The stresses are calculated from the stress-strain relations given in equations (7) to (9).

Piston Theory for Unsteady Aerodynamics in Flutter Analysis

When the expression for $W(x,y,\tau)$ given in equation (1) is substituted into equation (10) the following equation results:

$$p(x,y,\tau) = \left[m\omega^2 w - 2\rho a\left(1 + \frac{\gamma + 1}{4} M \frac{\partial h}{\partial x}\right)\left(i\omega w + v \frac{\partial w}{\partial x}\right)\right] e^{i\omega t} \tag{B15}$$

where the positive direction is upward and

m	mass of wing per unit area
ρ	mass density of air
a	speed of sound
γ	ratio of specific heats

M Mach number, v/a

h total local depth of wing

v velocity

As noted in the body of the report (see eq. (11)), the unsteady loading $p(x,y,\tau)$ can be taken to the left side of the equilibrium equation (eq. (3) or (B1)), thereby allowing the equation to be written in the form

$$\left\{\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} + \epsilon^4 \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \\ B_{20} & B_{21} & B_{22} \end{bmatrix}\right\} \begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{bmatrix} = 0 \tag{B16}$$

in which the components for A_{ij} are given in equation (B2) and the components for B_{ij} are as follows:

$$\left.\begin{aligned}
\left[B_{00}\right] &= -\omega^2\left[S_0\right] + 2\rho a i\omega\left(\left[F_0\right] + \frac{v(\gamma + 1)}{4a}\left[H_0\right]\right) \\
\left[B_{01}\right] &= -\omega^2\left[S_1\right] + 2\rho a\left(i\omega\left[F_1\right] + v\left[F_0\right]\right) + \frac{2\rho v(\gamma + 1)}{4}\left(i\omega\left[H_1\right] + v\left[H_0\right]\right) \\
\left[B_{02}\right] &= -\omega^2\left[S_2\right] + 2\rho a\left(i\omega\left[F_2\right] + v\left[F_1\right]\right) + \frac{2\rho v(\gamma + 1)}{4}\left(i\omega\left[H_2\right] + v\left[H_1\right]\right) \\
\left[B_{10}\right] &= -\omega^2\left[S_1\right] + 2\rho a i\omega\left(\left[F_1\right] + \frac{v(\gamma + 1)}{4a}\left[H_1\right]\right) \\
\left[B_{11}\right] &= -\omega^2\left[S_2\right] + 2\rho a\left(i\omega\left[F_2\right] + v\left[F_1\right]\right) + \frac{2\rho v(\gamma + 1)}{4}\left(i\omega\left[H_2\right] + v\left[H_1\right]\right) \\
\left[B_{12}\right] &= -\omega^2\left[S_3\right] + 2\rho a\left(i\omega\left[F_3\right] + 2v\left[F_2\right]\right) + \frac{2\rho v(\gamma + 1)}{4}\left(i\omega\left[H_3\right] + 2v\left[H_2\right]\right) \\
\left[B_{20}\right] &= -\omega^2\left[S_2\right] + 2\rho a i\omega\left(\left[F_2\right] + \frac{v(\gamma + 1)}{4a}\left[H_2\right]\right) \\
\left[B_{21}\right] &= -\omega^2\left[S_3\right] + 2\rho a\left(i\omega\left[F_3\right] + v\left[F_2\right]\right) + \frac{2\rho v(\gamma + 1)}{4}\left(i\omega\left[H_3\right] + v\left[H_2\right]\right) \\
\left[B_{22}\right] &= -\omega^2\left[S_4\right] + 2\rho a\left(i\omega\left[F_4\right] + 2v\left[F_3\right]\right) + \frac{2\rho v(\gamma + 1)}{4}\left(i\omega\left[H_4\right] + 2v\left[H_3\right]\right)
\end{aligned}\right\} \tag{B17}$$

$$\underset{k=0,1,2,3,4}{\left[S_k\right]} = \begin{bmatrix} S_{k,1} & & & & & \\ & S_{k,2} & & & & \\ & & \cdots & & & \\ & & & S_{k,N-1} & & \\ & & & & \frac{1}{2} S_{k,N} & \\ & & & & & 0 \end{bmatrix} \qquad \begin{matrix} \text{(B18)} \\ (N+1)\times(N+1) \end{matrix}$$

$$\underset{k=0,1,2,3,4}{\left[F_k\right]} = \begin{bmatrix} F_{k,1} & & & & & \\ & F_{k,2} & & & & \\ & & \cdots & & & \\ & & & F_{k,N-1} & & \\ & & & & \frac{1}{2} F_{k,N} & \\ & & & & & 0 \end{bmatrix} \qquad \begin{matrix} \text{(B19)} \\ (N+1)\times(N+1) \end{matrix}$$

$$\underset{k=0,1,2,3,4}{\left[H_k\right]} = \begin{bmatrix} H_{k,1} & & & & & \\ & H_{k,2} & & & & \\ & & \cdots & & & \\ & & & H_{k,N-1} & & \\ & & & & \frac{1}{2} H_{k,N} & \\ & & & & & 0 \end{bmatrix} \qquad \begin{matrix} \text{(B20)} \\ (N+1)\times(N+1) \end{matrix}$$

$$S_k = \int_{x_L(y)}^{x_T(y)} mx^k \, dx \qquad \text{(B21)}$$

$$F_k = \int_{x_L(y)}^{x_T(y)} x^k \, dx \qquad \text{(B22)}$$

$$H_k = \int_{x_L(y)}^{x_T(y)} \frac{\partial h}{\partial x} x^k \, dx \qquad \text{(B23)}$$

This completes the definition of terms that appear in equation (B16). The determinant of the matrix of coefficients of φ_i in equation (B16) is the flutter determinant (eq. (12)).

Mathematical Programming Techniques for Determining Minimum-Weight Design

In the interior penalty-function method denoted as "SUMT," the vector of design variables corresponding to the minimum value of the objective function is found by carrying out a sequence of minimizations of the P function. The P function $P(\vec{C},r_k)$ is defined as

$$P(\vec{C},r_k) = f(\vec{C}) + r_k \sum_j \frac{1}{g_j(\vec{C})} \tag{B24}$$

In equation (B24), $f(\vec{C})$ is the objective function, $\vec{C}$ is the vector of design variables, and $r_k \sum_j \frac{1}{g_j(\vec{C})}$ is the penalty term in which $g_j(\vec{C}) > 0$ are the inequality constraints and r_k is the constraint weighting factor. The subscript k placed on r refers to the sequence of values that r assumes during the minimization process. Let $\vec{C}_1$ be the vector of design variables which satisfies the inequality constraints and minimizes the P function for an initial value of r. Let that value of r be denoted as r_1. When $\vec{C}_1$ is found, the value of r is reduced to $r_2 < r_1$ and a search is carried out to find $\vec{C}_2$, which is the vector of design variables that minimizes $P(\vec{C},r_2)$. The starting point for that search is $\vec{C}_1$. When $\vec{C}_2$ is found, the value of r is reduced to $r_3 < r_2 < r_1$. Starting with $\vec{C}_2$, a search is carried out for $\vec{C}_3$. This sequence of searches is continued until the penalty term contributes little to the P function which is then approximately equal to the objective function.

The manner in which the P function varies with r is illustrated in figure 10 for a simple one-dimensional design space. The single design variable is x, the objective function is f(x), and the single inequality constraint is $g_1 \equiv x - b > 0$. The minimum value of P for $r = r_1$ is M_1; the minimum value of P for $r = r_2$ is M_2, and so forth. Figure 10 is based upon figure 4(a) in reference 12.

For each value of r an unconstrained minimization is carried out. This procedure involves several two-step searches. First, it is necessary to generate a set of direction cosines $\vec{Z}$ which defines a move direction in design-variable space. Second, a one-dimensional search is carried out to find the minimum value of P in the $\vec{Z}$ direction.

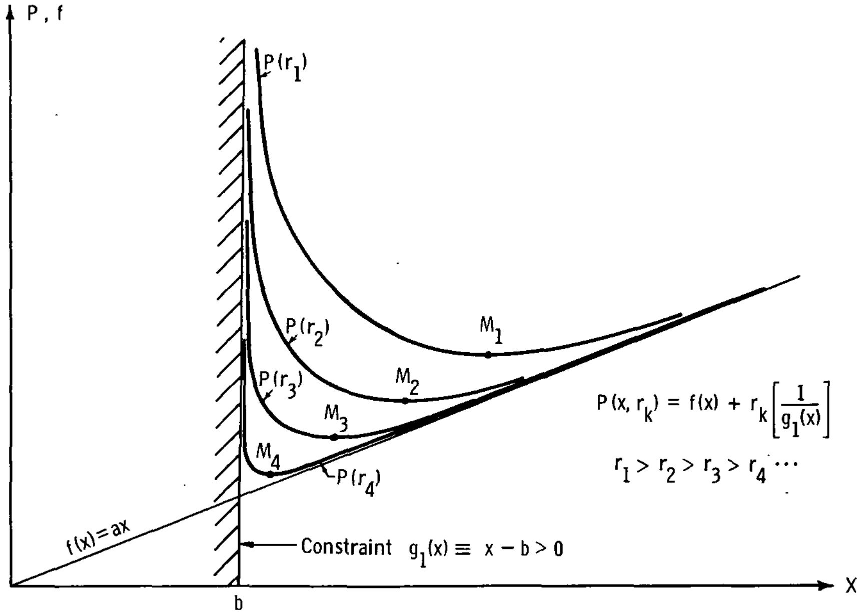


Figure 10.- Variation of P function with x for several values of the constraint weighting factor r. The minimum value of $P(r_k)$ is denoted by M_k. This is based upon figure 4(a) in reference 12.

In the present study, the direction-generating algorithm which was used almost exclusively was the Davidon-Fletcher-Powell method. Once a set of direction cosines $\vec{Z}$ has been found, the design variables $\vec{C}$ can be expressed as

$$\vec{C} = \vec{C}^* + S \cdot \vec{Z} \tag{B25}$$

in which S is the distance traveled in the $\vec{Z}$ direction and $\vec{C}^*$ is the vector of design variables for $S = 0$. The search in the $\vec{Z}$ direction is called a one-dimensional search because S is the only unknown.

In the one-dimensional search used to obtain the present results, the constraints $g_j(\vec{C})$ are approximated by polynomials in S. The polynomials never exceed second degree. Such a procedure is described in references 19 and 20. The flutter speed was found to be especially well-suited for polynomial approximation.

A more complete discussion of these optimization techniques is contained in references 12 and 17 to 23.

APPENDIX C

DESCRIPTION OF COMPUTER PROGRAM

The computer program SWIFT is written in FORTRAN IV for the CDC 6000 series computers at the Langley Research Center. The program also makes use of CDC library subroutine MATRIX (ref. 24). For these reasons the unmodified program may not be operational on other computer systems. A listing of the computer program is included so that mathematical operations can be examined in detail and so that modified programs can be developed. In the latter case this appendix can be considered to be an abbreviated user's document. It is assumed that the user is familiar with the material contained in the preceding sections of this report. A flow chart of the program is presented in figure 11.

SWIFT can be used to analyze or design a cantilever plate-like wing. The program calculates stress and deflections, natural frequencies, and flutter speeds and frequencies. When used in the design mode, the program can be used to find the minimum-weight wing cover-thickness distribution which satisfies flutter, strength, and minimum-gage design requirements. A six-design variable-strength design requires 4 to 6 minutes central-processing-unit execution (CPU) time. A six-design variable-flutter design requires 5 to 7 minutes CPU time. A strength-flutter (combined) design requires 6 to 8 minutes CPU time.

Description of Input

All of the input data are described in the main program. Most of the data are in namelist form. Additional description of input is given as follows:

The first card contains an optional word that is printed following the words "WING MATERIAL" in the output. "TITANIUM" is used in the sample problem in this appendix.

IOPT=1 — This option is used to find a value of OMEGA, the flutter frequency, and V, the flutter speed, in the neighborhood of the flutter solution. Real and imaginary values of the complex determinant are calculated for values of OMEGA and V starting with OMINIT and VINIT and ending with OMFIN and VFIN in increments of DELOM and DELV. A solution occurs when both the real and imaginary parts of the flutter determinant are zero simultaneously. Natural frequencies can be calculated by setting RHOA, the air density, equal to zero. The effect of aerodynamic damping on the vibration frequencies can be explored by setting only V equal to zero.

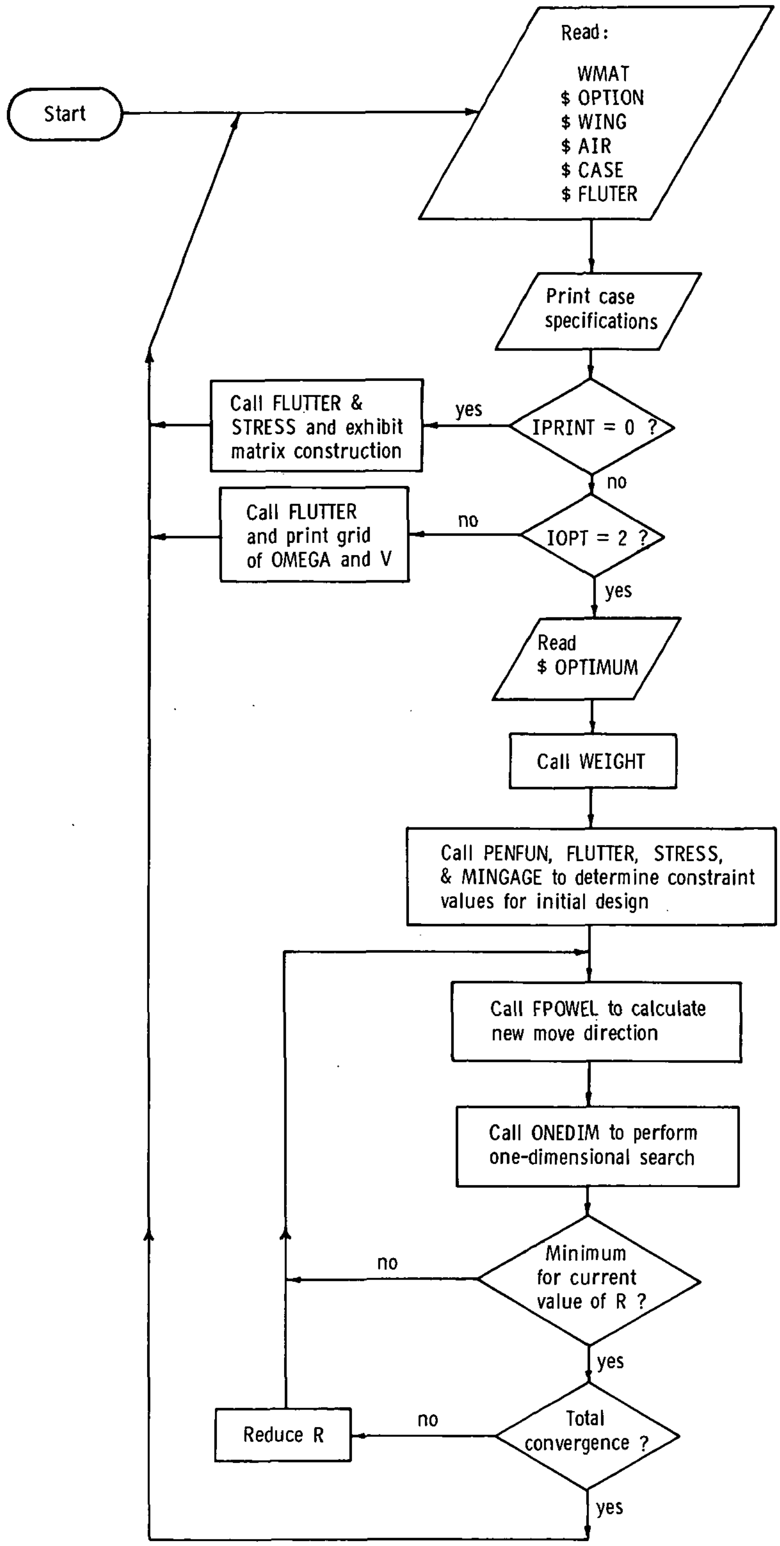


Figure 11.- Simplified flow chart of computer program SWIFT.

IOPT=2	Once the flutter-solution neighborhood has been established with IOPT=1, the program can be rerun with IOPT=2 to calculate flutter speed and frequency very accurately (within the limits of the theory) and to perform an optimization.
IPRINT=0	This option shows a printout of matrix construction. Matrix print statements require that NN, the number of spanwise stations, be set equal to 4. Final results can be used in deflection analysis.
IPRINT=1,2, or 3	These print options are used in the optimization mode.
ITHICK=2	The ITHICK=2 option is considered to give a more practical thickness distribution than the ITHICK=1 option. It is suggested that C(1) be set equal to the minimum-gage requirement. Limit active design variables to C(10), C(11), C(12), C(14), C(15), and C(16). Active design variables are those variables that are adjusted during the design process and are identified by nonzero DC(I). Set C(2) to C(9) and C(13) equal to zero.
DC(I)	These quantities are finite difference increments used to calculate derivatives with respect to the design variables C(I). If it is desired to hold a design variable fixed throughout the design process, set the DC(I) corresponding to that C(I) equal to zero.
NC	Subscript of the C(I) having the largest subscript. For example, if C(1), C(5), C(6), and C(14) are used to describe the thickness, then NC = 14.
IHP=0	Indicates that wing-depth-variation effects (sometimes called thickness effects) are not accounted for in piston theory.
IHP=1	Indicates that wing-depth-variation effects are accounted for in piston theory.
NN	Number of semispan stations, not counting the root station, where 10 is maximum and 6 to 8 appear to be adequate.
NCS	Number of chordwise stations used in numerical integration divided by 10. For example, when NCS=1, 10 stations are used by subroutine MGAUSS.
DELOM, DELV	Increments for OMEGA and V. Although these increments can be large for IOPT=1, they must be small for IOPT=2 where they are used as finite difference increments in a Newton-Raphson search to converge on OMEGA and V. For example, DELOM=1 and DELV=1000 are reasonable numbers for the IOPT=1 option.

However, for the IOPT=2 option these increments must be reduced to the order of DELOM=0.0001 and DELV=0.1.

R — Constraint weighting factor used in penalty-function formulation. Choose initial R so that the initial value of the penalty function is of the same order of magnitude as the objective function, which is the wing weight. In the program, both the stress and minimum-gage constraints have a built-in additional multiplier of 30. This additional multiplier was inserted because both the stress and minimum-gage constraints are averaged over the surface.

RDC — Factor by which R is divided at the end of each unconstrained search cycle. Numbers on the order of 5 to 15 are reasonable. The larger the value of RDC, the fewer the number of unconstrained search cycles. However, large values of RDC may also lead to false indications of convergence. Adjust RDC so that at least three unconstrained searches are carried out.

DESCON — Three logical variables. The values of each are T or F indicating true or false. The three variables refer to the three constraints being considered: flutter, stress, and minimum gage. For example, TFT indicates that the user wishes to consider the flutter and minimum-gage constraints but wishes to ignore the stress constraint. When a constraint is ignored, no analysis is performed to check that constraint.

Operational Hints

Flutter.- There are many values of OMEGA and V which satisfy the equations obtained by setting the flutter determinant equal to zero. During a computer run in the optimization mode IOPT=2, the program locks in on a flutter solution and follows it throughout the optimization process. To obtain the initial solution – that is, OMIN and VIN – the user runs the IOPT=1 option and chooses the solution which has the lowest flutter speed. For the final design, however, the program may not be examining the flutter solution corresponding to the lowest flutter speed.

The only way to guarantee that the final design does satisfy the flutter constraint is to rerun the IOPT=1 option using the final values of C(I). Although experience indicates that this procedure is seldom required, it is recommended until a user becomes familiar with the particular class of problems on which he is working.

Convergence.- Convergence to a global extremum is of prime concern to the user. In general, the best way to increase one's confidence in a particular solution is to rerun the problem with a different initial design, different initial value of R, and/or a different

value of RDC. A technique is presented in reference 25 for warping design variable space in such a way that the second run will have a greater chance of locating another minimum, if one exists.

The smaller the final values of a constraint function G(I), the closer the design is to that constraint. The following values of the constraint functions G(I) are reasonable for convergence: G(1) = 0.03, G(2) = 0.15, and G(3) = 0.15. Designs having a value of G(I) larger than those indicated should be examined for false convergence.

Scaling of flutter determinant.- The object of subroutine DEUPPS is to evaluate the complex flutter determinant as rapidly as possible. Because of the overriding need for speed, it was decided to eliminate certain scaling logic that would normally prevent overflow and underflow. Scaling must be carried out by the user. The elements in the complex flutter matrix are scaled in statements 101, 102, 103, and 104 in subroutine FLUTTER.

Program listing and sample output.- The remaining part of this appendix contains a listing of the program denoted as "SWIFT," a printout of a computer run used to determine the flutter-solution neighborhood, and a printout of an optimization run.

```
      PROGRAM BIGBOY(INPUT=201,OUTPUT=201,TAPE5=INPUT)

C   THIS IS THE MAIN PROGRAM.
C   ITS FUNCTION IS TO READ AND PRINT MOST OF THE INPUT VARIABLES, SET
C        VALUES FOR A NUMBER OF CONSTANTS, AND INITIATE THE CALLING OF
C        OTHER SUBROUTINES IN AN APPROPRIATE SEQUENCE AS SPECIFIED BY
C        THE INPUT OPTIONS.

    1 FORMAT(1H1    /5X*DETERMINATION OF THICKNESS DISTRIBUTION FOR MINI
     1MUM WEIGHT WING*/
     25X*SATISFYING PRESCRIBED FLUTTER AND STRENGTH CRITERIA.*/
     35X*REFERENCE - NACA TN-3640,    STEIN-SANDERS,    JUNE,1956.*/
     45X*DEXTER-STROUD     RDK288     MAY,1969.*///
     55X*DATE*A10)
  100 FORMAT(A10)
  502 FORMAT(/103X*XXX*/103X*XXX*/* WING DESCRIPTION*,86X*XXX*,
     110X*FIXED BOUNDARY*/102X*.XXX*,10X*CONDITION*/
     15X*ASPECT RATIO.............AR =*E16.8,52X
     1*. XXX*/
     25X*SURFACE AREA...........AREA =*E16.8,7H IN.**2,44X
     2*.  XXX*/
     35X*L.E. SWEEP ANGLE.......ANG =*E16.8* DEGREES*,42X
     3*.   XXX*/
     45X*SEMISPAN...............SS =*E16.8* IN.*,45X
     4*.    XXX*/
     55X*ROOT CHORD.............RC =*E16.8* IN.*,44X
     5*.     XXX*)
  512 FORMAT(5X*DEPTH RATIO.............DR =*E16.8,47X
     6*.      XXX*/
     75X*WING MATERIAL.............*A10,54X
     7*.       XXX  ROOT CHORD =*F6.1* IN.*/
     85X*MASS DENSITY..........RHOW =*E16.8,23H (LB.*SEC.**2)/(IN.**4)
     822X
     8*.        XXX*/
     95X*YOUNGS MODULUS...........E =*E16.8* PSI*,40X
     9*.         XXX*/
     15X*PCISSONS RATIO..........MU =*E16.8,43X
     1*.          XXX*/
     291X*.            XXX*/90X*.*F3.0* DEG.    XXX*)
  503 FORMAT(//* EQUATION FOR THICKNESS - TBAR    (INCHES)*/)
 5031 FORMAT(* TBAR =*3X41HC( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3/
     18X46H+(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B/
```

```
   18X49H+(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*B**2/
   18X49H+(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*B**3//
   112X43HWHERE  A = 1-Y/SS   AND   B = 1-2*X/(LE+TE)//
   126X12H(ITHICK = 1))
5032 FORMAT(* TBAR =*3X41HC( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3/
   18X46H+(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B/
   18X52H+(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*(1-B*B)/
   18X54H+(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*(1-B*B)*B//
   112X43HWHERE  A = 1-Y/SS   AND   B = 1-2*X/(LE+TE)//
   126X12H(ITHICK = 2))
 531 FORMAT(//5X*DESIGN VARIABLES*,17X*FINITE DIFFERENCE INCREMENTS*,
   16X*ANALYTICAL DERIVATIVES*
   2 11X1H*,* INDICATES ACTIVE*/107X*DESIGN VARIABLES*)
 533 FORMAT(5X*C(*I2*) =*E16.8,10X*CC(*I2*) =*E16.8,10X*AD(*I2*) =*E16.
   18)
 534 FORMAT(2X1H*,2X*C(*I2*) =*E16.8,10X*DC(*I2*) =*E16.8,10X*AD(*I2*)
   1=*E16.8)
 504 FORMAT(* AIR PROPERTIES*75X
   1*. . . . . . .XXX*/103X*XXX*/
   15X*SPEED OF SOUND...........A =*E16.8* IN./SEC. =*E16.8* FT./SEC.*
   2 5X*SEMISPAN =   XXX*,21H LEADING EDGE = A+B*Y/
   25X*MASS DENSITY..........RHOA =*E16.8,23H (LB.*SEC.**2)/(IN.**4),
   218XF5.1,* IN.    XXX       A =*E16.8/
   35X*CORRESPONDING ALTITUDE.... =*E16.8* FT.*60X
   3*B =*E16.8)
  5 FORMAT(//* STATIONS*//
   15X*NUMBER OF SEMISPAN STATIONS.................NN =*I3/
   25X*NUMBER OF CHORDWISE STATIONS = (10 X NCS)...NCS =*I3)
 22 FORMAT(//* RANGE AND INCREMENT OF ITERATION*//
   15X*OMEGA INITIAL.............OMINIT =*E16.8* RADIANS/SEC.*/
   25X*OMEGA INCREMENT...........DELOM =*E16.8* RADIANS/SEC.*/
   35X*OMEGA FINAL...............CMFIN =*E16.8* RADIANS/SEC.*//
   45X*V INITIAL.................VINIT =*E16.8* IN./SEC.*/
   55X*V INCREMENT...............CELV =*E16.8* IN./SEC.*/
   65X*V FINAL...................VFIN =*E16.8* IN./SEC.*)
 50 FORMAT(//* INPUT OPTIONS*//
   15X*   IOPT =*I2,5X*1 - ITERATE ON OMEGA AND V, 2 - COMPLEX SEARCH*
   1/
   15X* IPRINT =*I2,5X*0 - MATRIX CONSTRUCTION, 1 - SHORT PRINTOUT, 2
   1- INTERMEDIATE PRINTOUT, 3 - EXTENDED PRINTOUT*/
   75X*ISNDWCH =*I2,5X*0 - SOLID WING, 1 - SANDWICH CONSTRUCTION*/
   85X*    IHP =*I2,5X*0 - THICKNESS VARIATION EFFECTS NOT ACCOUNTED F
   8OR IN PISTON THEORY, 1 - THICKNESS EFFECTS ARE USED*/
```

```
     8)

      REAL MU
      DIMENSION RESULT(2)
      DIMENSICN H(16)
      COMPLEX AAC
      COMMCN
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     1/BLK20/IFIRST,IFAIL,NOCONV,NOS
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     4/BLK4/A,RHOA,IHP
     5/BLK5/OMEGA1,DELOM,OMFIN,V1,DELV,VFIN
     1/BLK6/IOPT,IPRINT,ISNDWCH
     1/BLK12/VCR,R,NC,H,DESCON(3)
     3/BLK13/Y,C1,C2
     7/BLK17/NEWR
     1/BLK50/RHOF
     8/BLK18/DR
     1/BLK28/CSAV(16),DC(16),D(16),DIRCOS(16),DM1,X
     2/BLK32/IG
     3/BLK36/C1A,C1B,C2A
     3/BLK40/AD(16)
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLKNPS/AREA
     1/NPSBLK/ITHICK
      NAMELIST/OPTION/IOPT,IPRINT,ISNDWCH,ITHICK
      NAMELIST/WING/ANG,RHOW,E,MU,WTF,AR,AREA,DR,NC,C,DC
      NAMELIST/AIR/A,RHOA,CORALT,IHP
      NAMELIST/CASE/NN,NCS
      NAMELIST/FLUTER/OMINIT,DELCM,OMFIN,VINIT,DELV,VFIN



C        INPUT DATA CARDS


C   1 - WING MATERIAL   (FORMAT A10)

C   2 - $OPTICN
C              IOPT    1-ITERATE CMEGA AND V, 2-COMPLEX SEARCH,OPTIMIZATICN
C            IPRINT    0-MATRIX CCNSTRUCTION - FOR TROUBLE-SHOOTING,
C                            USE NN=4 ONLY
C                      1-SHORT PRINTOUT - INITIAL CASE PLUS FINAL RESULT
```

```
C                      2-INTERMEDIATE PRINTOUT - FINAL RESULTS OF
C                           FLETCHER-POWELL AND ONE-DIMENSIONAL SEARCH
C                      3-EXTENDED PRINTOUT - FOR TROUBLE-SHOOTING.
C                           GIVES FINITE DIFFERENCE AND ONE-DIMENSIONAL
C                           SEARCH DETAILS
C             ISNDWCH  0-SOLID WING, 1-SANDWICH CONSTRUCTION
C              ITHICK  1-GENERAL 16-TERM THICKNESS EQUATION
C                      2-MODIFIED THICKNESS EQUATION - THICKNESS IS
C                           EQUAL TO C(1) ALONG LEADING EDGE, TRAILING
C                           EDGE, AND TIP

C   3 - $WING
C                 ANG  LEADING EDGE SWEEP ANGLE  -  DEGREES
C                RHOW  MASS DENSITY  -  LB.*SEC.**2/IN.**4
C                   E  YOUNGS MODULUS  -  PSI
C                  MU  POISSONS RATIO
C                 WTF  WEIGHT OF FUEL IN ONE WING  -  POUNDS
C                  AR  ASPECT RATIO - SPAN**2/TOTAL PLANFORM SURFACE AREA
C                AREA  SURFACE AREA OF SEMISPAN  -  IN.**2
C                  DR  DEPTH RATIO - RATIO OF MAXIMUM DEPTH OF AIRFOIL TO
C                               LOCAL CHORD
C                  NC  NUMBER OF DESIGN VARIABLES   (16 MAXIMUM)
C                   C  DESIGN VARIABLES   (16 MAXIMUM)
C                  DC  FINITE DIFFERENCE INCREMENTS   (16 MAXIMUM)

C   4 - $AIR
C                  A  SPEED OF SOUND  -  INCHES/SECOND
C               RHOA  MASS DENSITY  -  LB.*SEC.**2/IN.**4
C             CORALT  CORRESPONDING ALTITUDE  -  FEET
C                IHP  0-NO HPRIME TERMS IN COMPLEX MATRIX, 1-HPRIME TERMS

C   5 - $CASE
C                 NN  NUMBER OF SEMISPAN STATIONS   (10 MAXIMUM)
C                NCS  NUMBER OF CHORDWISE STATIONS/10

C   6 - $FLUTER
C             OMINIT  OMEGA INITIAL  -  RADIANS/SECOND
C              DELOM  OMEGA INCREMENT  -  RADIANS/SECOND
C              OMFIN  OMEGA FINAL (IF IOPT=1)
C              VINIT  V INITIAL  -  INCHES/SECOND
C               DELV  V INCREMENT  -  INCHES/SECOND
C               VFIN  V FINAL (IF IOPT=1)
```

```
C   FOLLOWING CARDS NECESSARY CNLY IF IOPT=2, BUT MAY BE LEFT IN IF IOPT=1

C    7 - $OPTIMUM
C                R   FACTOR IN PENALTY FUNCTION
C              RDC   R REDUCTION FACTOR
C              VCR   V CRITICAL  -  INCHES/SECOND

C    8 - DESCON   3 DESIGN CONSTRAINTS CONSIDERED, T OR F   (IN ORDER FLUTTER,
C                      STRESS, MINGAGE)      (FORMAT 3L5)

C    9 - $STRSS      (NECESSARY ONLY IF DESCON(2)=T)
C           PLOAD    UNIFORM PRESSURE LOADING  -  PSI
C             YCR    YIELD STRESS  -  PSI

C   10 - $GAGE       (NECESSARY CNLY IF DESCON(3)=T)
C           TBARMIN   MINIMUM GAGE  -  INCHES



      CALL DAYTIM(RESULT)
 1000 CONTINUE
      READ 100, WMAT
      IF(EOF,5) 1001,1002
 1001 STOP
 1002 CONTINUE
      READ OPTION
      READ WING
      READ AIR
      READ CASE
      READ FLUTER
      IG = 0
      PI = 3.14159265359
      AFT = A/12.
      OMEGA1 = OMINIT
      V1 = VINIT
      SS = SQRT(AR*AREA/2.)
      C2A = 0.
      C1B = TAN(ANG*PI/180.)
      RC = .5*SS*C1B+AREA/SS
      C1A = C2A-RC
      EPS = SS/FLOAT(NN)
      MCS=11
      NN1 = NN+1
```

```
      NN2 = NN+2
      NN22 = 2*NN+2
      NN23 = 2*NN+3
      NN33=3*NN+3
      CALL ANDER
      PRINT 1, RESULT(1)
      PRINT 502, AR,AREA,ANG,SS,RC
      PRINT 512, DR,WMAT,RC,RHOW,E,MU,ANG
      PRINT 504, A,AFT,RHOA,SS,C1A,CORALT,C1B
      IF(ISNDWCH.EQ.0) GO TO 400
      Y = 0.
      CALL CEE
      A1 = 2./3.*DR*(C2-C1)**2
      Y = SS
      CALL CEE
      A2 = 2./3.*DR*(C2-C1)**2
      VOL = 1./3.*(A1+A2+SQRT(A1*A2))*SS
      RHOF = WTF/(VOL*386.)
      PRINT 505, WTF,VOL,RHOF
  505 FORMAT(//* FUEL PROPERTIES*//
     15X*WEIGHT OF FUEL...........WTF =*E16.8* LB.*/
     25X*VOLUME OF WING...........VOL =*E16.8,7H IN.**3/
     35X*DENSITY OF FUEL.........RHOF =*E16.8,23H (LB.*SEC.**2)/(IN.**4)/
     1)
  400 CONTINUE
      PRINT 503
      IF(ITHICK.EQ.1) PRINT 5031
      IF(ITHICK.EQ.2) PRINT 5032
      PRINT 531
      DO 532 I=1,NC
      IF(DC(I).EQ.0.) GO TO 535
      PRINT 534, I,C(I),I,DC(I),I,AD(I)
      GO TO 532
  535 CONTINUE
      PRINT 533, I,C(I),I,DC(I),I,AD(I)
  532 CONTINUE
  600 CONTINUE
      PRINT 5, NN,NCS
      PRINT 50, IOPT,IPRINT,ISNDWCH,IHP
      CALL FMAT
      CALL HMAT
      CALL WEIGHT(WT)
      PRINT 601, WT
```

```
601 FORMAT(//////* INITIAL WEIGHT =*E16.8,*  POUNDS*//)                    25800000
    IF(IPRINT.EQ.0) GO TO 201                                              25900000
    IF(IOPT.EQ.2) GO TO 202                                                26000000
    NOCONV = 0                                                             26100000
    NEWR = 1                                                               26200000
    PRINT 22, OMINIT,DELCM,OMFIN,VINIT,DELV,VFIN                           26300000
    CALL FLUTTER                                                           26400000
    GO TO 1000                                                             26500000
201 CONTINUE                                                               26600000
    NOCONV = 0                                                             26700000
    NEWR = 0                                                               26800000
    CALL FLUTTER                                                           26900000
    CALL STRESS(G2)                                                        27000000
    GO TO 1000                                                             27100000
202 CONTINUE                                                               27200000
    CALL METHOD                                                            27300000
    GO TO 1000                                                             27400000
    END                                                                    27500000
```

```
      SUBROUTINE MGAUSS(A,B,N,SUM,FUNC,FOFX,NUMBER)
********* DOCUMENT DATE 08-01-68   SUBROUTINE REVISED 08-01-68 *********D1.1   2
C         THIS SUBROUTINE INTEGRATES FROM ZERO TO ONE
      DIMENSION U(5),R(5),SUM(1),FOFX(1)
      COMMON
     2/BLK20/IFIRST,IFAIL,NOCONV
      DO 1 LL=1,NUMBER
    1 SUM(LL)=0.0
      IF(A.EQ.B)RETURN
      U(1)=.425562830509184
      U(2)=.283302302935376
      U(3)=.160295215850488
      U(4)=.067468316655508
      U(5)=.013046735741414
      R(1)=.147762112357376
      R(2)=.134633359654998
      R(3)=.109543181257991
      R(4)=.074725674575290
      R(5)=.033335672154344
      FINE= N
      DELTA=FINE/(B-A)
      DO 3 K=1,N
      XI=K-1
      FINE=A+XI/DELTA
      DO 2 II= 1,5
      UU=U(II)/DELTA+FINE
      CALL FUNC (UU,FOFX)
      IF(IFAIL.EQ.1) RETURN
      DO 2 JOYBOY=1,NUMBER
    2 SUM(JOYBOY)=R(II)*FOFX(JOYBOY)+ SUM(JOYBOY)
      DO 3 JJ=1,5
      UU=(1.0-U(JJ))/DELTA+FINE
      CALL FUNC (UU,FOFX)
      IF(IFAIL.EQ.1) RETURN
      DO 3 NN=1,NUMBER
    3 SUM(NN)=R(JJ)*FOFX(NN)+SUM(NN)
      DO 7 IJK= 1,NUMBER
    7 SUM(IJK)=SUM(IJK)/DELTA
      RETURN
      END
```

```
      SUBROUTINE ANDER

C   ANALYTICAL DERIVATIVES OF WEIGHT WITH RESPECT TO DESIGN VARIABLES
C        C(I)

      REAL MU
      COMMON
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     3/BLK36/C1A,C1B,C2A
     3/BLK40/AD(16)
     1/NPSBLK/ITHICK
      SIXINV = 1./6.
      THREINV = 2.*SIXINV
      RHO2 = RHOW*772.
      CCS = (C2A-C1A)*SS
      CBS = C1B*SS**2
      DO 100 I=1,16
  100 AD(I) = 0.
      AD(1) = RHO2*(CCS-.5*CBS)
      AD(2) = RHO2*(.5*CCS-SIXINV*CBS)
      AD(3) = RHO2*(THREINV*CCS-.5*SIXINV*CBS)
      AD(4) = RHO2*(.25*CCS-.05*CBS)
C   FOLLOWING 4 ANALYTIC DERIVATIVES CORRECT ONLY FOR C2=TE=0
      AD(9) = THREINV*AD(1)
      AD(10) = THREINV*AD(2)
      AD(11) = THREINV*AD(3)
      AD(12) = THREINV*AD(4)
      IF(ITHICK.EQ.1) RETURN
      AD(10) = AD(2) - AD(10)
      AD(11) = AD(3) - AD(11)
      AD(12) = AD(4) - AD(12)
      RETURN
      END
```

```
      SUBROUTINE METHOD

   10 FORMAT(///*    WT = WEIGHT IN POUNDS =*E16.8/*    PF = PENALTY FUN
     1CTION =*E16.8/* PFUNC =      WT + PF     =*E16.8//* CONSTANTS 1 -*
     1I3/(4E20.8))
   15 FORMAT(/125(1H*)/)
   18 FORMAT(//* INPUT DATA FOR SEARCH ROUTINE*//
     15X*NUMBER OF DESIGN VARIABLES....NC =*I3/
     15X*                              R =*E16.8/
     15X*R REDUCTION FACTOR...........RDC =*E16.8)
   17 FORMAT(//* INPUT DATA FOR STRESS*//
     35X*UNIFORM PRESSURE LOADING...PLOAD =*E16.8* PSI*/
     45X*YIELD STRESS.................YCR =*E16.8* PSI*)
   19 FORMAT(//* INPUT FOR MINGAGE*//
     15X*MINIMUM GAGE.............TBARMIN=*E16.8* INCHES*)
   20 FORMAT(3L5)
   21 FORMAT(//* DESIGN CONSTRAINTS CONSIDERED*/
     15X*FLUTTER -*L2/5X* STRESS -*L2/5X*MINGAGE -*L2)
   28 FORMAT(//* INPUT DATA FOR FLUTTER*//
     15X*OMEGA INITIAL.............OMINIT =*E16.8* RADIANS/SEC.*/
     25X*OMEGA INCREMENT............DELOM =*E16.8* RADIANS/SEC.*//
     35X*V INITIAL..................VINIT =*E16.8* IN./SEC. =*E16.8* FT.
     3/SEC.*/
     45X*V INCREMENT.................DELV =*E16.8* IN./SEC. =*E16.8* FT.
     4/SEC.*//
     55X*V CRITICAL...................VCR =*E16.8* IN./SEC. =*E16.8* FT.
     5/SEC.*)
   27 FORMAT(/* FLETCHER - POWELL SEARCH ROUTINE*/
     1* PERTURBED CASES*)

      DIMENSION H(16)
      LOGICAL DESCON
      REAL MU
      COMMON
     2/BLK2/CMEGA,V
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     5/BLK5/CMEGA1,DELOM,OMFIN,V1,DELV,VFIN
     1/BLK6/ICPT,IPRINT,ISNDWCH
     8/BLK8/KDERIV
     1/BLK12/VCR,R,NC,H,DESCON(3)
     2/BLK20/IFIRST,IFAIL,NOCONV,NCS
     2/BLK14/YCR,IY
```

```
     1/BLK15/PLOAD
     7/BLK17/NEWR
     3/BLK25/TBARMIN
     1/BLK28/CSAV(16),DC(16),D(16),DIRCOS(16),DM1,PF
     3/BLK33/G1,G2,G3
     5/BLK35/OMEGAO,VO
     7/BLK37/G11,G22,G33
     3/BLK40/AD(16)
     4/BLK41/WT
     4/BLK44/RDC
      NAMELIST/OPTIMUM/RDC,R,VCR
      NAMELIST/STRSS/PLOAD,YCR
      NAMELIST/GAGE/TBARMIN
      READ OPTIMUM
      READ 20, (DESCON(J),J=1,3)
      IF(DESCON(2)) READ STRSS
      IF(DESCON(3)) READ GAGE
      PRINT 18, NC,R,RDC
      PRINT 21, (DESCON(J),J=1,3)
      V1F = V1/12.
      DELVF = DELV/12.
      VCRF = VCR/12.
      IF(DESCON(1)) PRINT 28, OMEGA1,DELOM,V1,V1F,DELV,DELVF,VCR,VCRF
      IF(DESCON(2)) PRINT 17, PLOAD,YCR
      IF(DESCON(3)) PRINT 19, TBARMIN
      DO 302 J=1,NC

C  EVALUATE NOMINAL WEIGHT

  302 CSAV(J) = C(J)
      IFAIL = 0
      NEWR = 1
      CALL WEIGHT(WT)

C  EVALUATE PF (PENALTY FUNCTION)

      KDERIV = 0
      IFIRST = 1
      CALL PENFUN(PF)
      IF(IFAIL.EQ.1) GO TO 550
      OMEGAO = OMEGA
      VO = V
      G11 = G1
```

```
      G22 = G2
      G33 = G3
      OMEGA1 = OMEGA
      V1 = V
      PFUNC=WT+PF
      PRINT 10, WT,PF,PFUNC,NC,(C(J),J=1,NC)
      DO 100 J=1,NC
  100 CSAV(J) = C(J)
      IFIRST = 2

      IF(IPRINT.EQ.3) PRINT 27
      CALL FPCWEL
      RETURN
  550 PRINT 50
   50 FORMAT(/* FAILURE WITH INITIAL DESIGN*)
      STOP
      END
```

```
      SUBROUTINE FPOWEL
C     DAVIDON-FLETCHER-POWELL DIRECTION GENERATOR PLUS CONVERGENCE TESTS

    1 FORMAT(/5X*PF =*E16.8)
    3 FORMAT(/* PF PLUS =*E16.8/(5E25.8))
    4 FORMAT(/* PF MINUS =*E16.8/(5E25.8))
   15 FORMAT(/125(1H*)/)
  111 FORMAT(125(1H*))
  110 FORMAT(/* MOVE DIRECTION DATA FOR ONE-DIMENSIONAL SEARCH*/)
   14 FORMAT(/5X*FIRST DERIVATIVES*/(4X8E16.8))
   17 FORMAT(/5X*DIRECTION COSINES*/(4X8E16.8))
   19 FORMAT(/5X*CSAV*/(4X8E16.8))
   18 FORMAT(/5X*PFUNC = WT+PF =*E16.8,10X*DERIVATIVE OF PFUNC IN MOVE D
     1IRECTION =*E16.8)
  109 FORMAT(//)
  112 FORMAT(/* ONE DIMENSIONAL SEARCH - FLETCHER-POWELL METHOD*/)
   33 FORMAT(/* DERIVATIVE OF PFUNC IN MOVE DIRECTION IS POSITIVE - REIN
     1ITIALIZE HH MATRIX*)
   36 FORMAT(/* DERIVATIVE OF PFUNC IN MOVE DIRECTION IS POSITIVE WITH I
     1NITIAL HH MATRIX - PROCEED TO CTEST*)
   62 FORMAT(/* GRADIENT CONVERGENCE TEST SATISFIED*)
   76 FORMAT(/* Q NEGATIVE - REINITIALIZE HH MATRIX*)
   81 FORMAT(/* CUADRATIC CONVERGENCE TEST SATISFIED*)
  101 FORMAT(/* MINIMUM MOVE SIZE CONVERGENCE TEST SATISFIED*)
  301 FORMAT(/* ESTIMATED MINIMUM WEIGHT LESS THAN 2 PERCENT LOWER THAN
     1PRESENT WEIGHT*/* DESIGN CONSIDERED CONVERGED*)
  311 FORMAT(/* CURRENT VALUE OF R IS*E16.8)
  905 FORMAT(/* PF =*E16.8,*   WT =*E16.8,*   S =*E16.8//* CONSTANTS*/
     1(4E16.8))
  921 FORMAT(/* CTEST =*E16.8)

      REAL MU,M1,I,IPLUS,IMINUS
      REAL M2,MPLUS
      LOGICAL DESCON
      DIMENSION H(16)
      DIMENSION SIG(16),Y(16),SIGT(16),YT(16),A(16,16),B(16,16),
     1HH(16,16),D2(16)
      COMMON
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     8/BLK8/KDERIV
```

```
     1/BLK12/VCR,R,NC,H,DESCON(3)
     7/BLK17/NEWR
     2/BLK20/IFIRST,IFAIL,NOCONV,NOS
     1/BLK28/CSAV(16),DC(16),D(16),DIRCOS(16),DM1,I
     1/BLK31/DCR,OMPLUS,VPLUS
     3/BLK33/G1,G2,G3
     4/BLK34/GP1,GP2,GP3
     3/BLK40/AD(16)
     4/BLK41/WT
     5/BLK42/M1
     1/BLK43/S
     4/BLK44/RDC
      EQUIVALENCE (I,PF),(SIGT,YT),(A,B)

      MULT = 20
      ISCLR = 24
      ITRANS = 0
      ISUB = 22
      IADD = 21

C   CENTRAL CIFFERENCE CALCULATICN OF FIRST DERIVATIVES

   10 CONTINUE
      KOUNTR = 0
      KDERIV = 1
      DO 11 K=1,NC
C   SET D = 0 FOR INACTIVE DESIGN VARIABLES
      IF(DC(K).EQ.0.) GO TO 12
      C(K) = CSAV(K)+DC(K)
      CALL PENFUN(IPLUS)
      IF(IFAIL.EQ.1) GO TO 501
      IF(IPRINT.NE.3) GO TO 200
      PRINT 3, IPLUS,(C(J),J=1,NC)
      PRINT 15
  200 CONTINUE
      C(K) = CSAV(K)-DC(K)
      CALL PENFUN(IMINUS)
      IF(IFAIL.EQ.1) GO TO 501
      IF(IPRINT.NE.3) GO TO 201
      PRINT 4, IMINUS,(C(J),J=1,NC)
      PRINT 15
  201 CONTINUE
      D(K) = (IPLUS-IMINUS)/(2.*DC(K))+AD(K)
```

```
      C(K) = CSAV(K)
      GO TO 11
   12 D(K) = 0.
   11 CONTINUE
      GO TO 20
  501 DO 511 J=1,NC
  511 DC(J) = DC(J)/2.
      IFAIL = 0
      GO TO 10

C     HH MATRIX INITIALIZED TO IDENTITY MATRIX

   20 CONTINUE
      NCA = NC
      DO 48 J=1,NC
      IF(DC(J).EQ.0.) NCA = NCA-1
   48 CONTINUE
      DO 21 K=1,NCA
      DO 21 J=1,NCA
      HH(J,K) = 0.
      IF(J.EQ.K) HH(J,K) = 1.
   21 CONTINUE
      KOUNT = 0
   49 CONTINUE

C     REDUCE D VECTOR TO NCA    (ACTIVE DESIGN VARIABLES ONLY)

      NCA = NC
      IF(NC.EQ.1) GO TO 30
      J = NC-1
   46 CONTINUE
      IF(DC(J).NE.0.) GO TO 37
      NCA = NCA-1
      DO 38 JJ=J,NCA
   38 D(JJ) = D(JJ+1)
   37 CONTINUE
      J = J-1
      IF(J.EQ.0) GO TO 47
      GO TO 46
   47 CONTINUE
      IF(DC(NC).EQ.0.) NCA = NCA-1
   30 KOUNT = KOUNT+1
      KOUNTR = KOUNTR+1
```

```
      SCLR = -1.
      CALL MATRIX(ISCLR,NCA,NCA,0,SCLR,0,HH,16,HH,16)
      CALL MATRIX(MULT,NCA,NCA,1,HH,16,D,16,H,16)
      CALL MATRIX(ISCLR,NCA,NCA,0,SCLR,0,HH,16,HH,16)

C     EXPAND D AND H VECTORS TO NC

      IF(NC.EC.1) GO TO 53
      K = NC-1
      DO 39 J=1,K
      IF(DC(J).NE.0.) GO TO 39
      J1 = NCA
      DO 45 JJ=J,NCA
      D(J1+1) = D(J1)
      H(J1+1) = H(J1)
      J1 = J1-1
   45 CONTINUE
      NCA = NCA+1
      D(J) = 0.
      H(J) = 0.
   39 CONTINUE
      IF(DC(NC).EQ.0.) D(NC) = H(NC) = 0.
   53 CONTINUE
      SUM = 0.
      DO 31 K=1,NC
   31 SUM = SUM+H(K)**2
      SUM = SQRT(SUM)
      DCR = 0.
      DO 32 K=1,NC
      DIRCOS(K) = H(K)/SUM
      IF(ABS(DIRCOS(K)).LT.DCR) GO TO 32
      KK = K
      DCR = ABS(DIRCOS(K))
   32 CONTINUE

C     DCR EQUALS THE FINITE DIFFERENCE INCREMENT FOR THE DIRECTION HAVING
C         THE LARGEST DIRECTION COSINE (DIRCOS(K))

      DCR = DC(KK)
   35 CONTINUE

C     SHORT STEP IN MOVE DIRECTION
```

```
      DO 34 K=1,NC
   34 C(K) = CSAV(K)+DIRCOS(K)*DCR
      KDERIV = 1
      IF(IPRINT.EQ.3) PRINT 2
    2 FORMAT(* SHORT STEP IN MOVE DIRECTION*)
      CALL PENFUN(IPLUS)
      IF(IFAIL.EQ.1) GO TO 503
      GP1 = G1
      GP2 = G2
      GP3 = G3
      IF(IPRINT.NE.3) GO TO 202
      PRINT 3, IPLUS,(C(J),J=1,NC)
      PRINT 17, (DIRCOS(J),J=1,NC)
  202 CONTINUE
      DM1 = (IPLUS-I)/DCR
      DO 321 K=1,NC
  321 DM1 = DM1+DIRCOS(K)*AD(K)
      IF(DM1.LT.0.) GO TO 41
      IF(IPRINT.EQ.3) PRINT 15
      IF(KOUNT.EQ.1) GO TO 322
      IF(IPRINT.GT.1) PRINT 33
      GO TO 20
  322 IF(IPRINT.GT.1) PRINT 36
      GO TO 300
  503 CONTINUE
      DCR = DCR/2.
      IFAIL = 0
      GO TO 35
   41 CONTINUE
      M1 = PF+WT
      IF(IPRINT.EQ.1) GO TO 431
      PRINT 109
      DO 42 INDEX=1,3
   42 PRINT 111
      PRINT 110
      PRINT 14, (D(J),J=1,NC)
      PRINT 19, (CSAV(J),J=1,NC)
      PRINT 17, (DIRCOS(J),J=1,NC)
      PRINT 18, M1,DM1
      PRINT 109
      DO 43 INDEX=1,3
   43 PRINT 111
      IF(IPRINT.EQ.3) PRINT 112
```

```
C     CALL ONE-DIMENSIONAL SEARCH

  431 CALL ONEDIM
      DO 44 K=1,NC
   44 C(K) = CSAV(K)
      IF(IPRINT.NE.1) PRINT 905, I,WT,S,(C(K),K=1,NC)
      IF(ABS(S-C.).LT.1.E-8) GO TO 20
   50 CONTINUE

C     CENTRAL DIFFERENCE CALCULATION OF FIRST DERIVATIVES

      IF(IPRINT.NE.3) GO TO 203
      PRINT 15
      PRINT 5
    5 FORMAT(/* FLETCHER - POWELL SEARCH ROUTINE*/* PERTURBED CASES*)
  203 CONTINUE
      KDERIV = 1
      DO 51 K=1,NC
C     SET D2 = 0 FOR INACTIVE DESIGN VARIABLES
      IF(DC(K).EQ.0.) GO TO 52
      C(K) = CSAV(K)+DC(K)
      CALL PENFUN(IPLUS)
      IF(IFAIL.EQ.1) GO TO 502
      IF(IPRINT.NE.3) GO TO 204
      PRINT 3, IPLUS,(C(J),J=1,NC)
      PRINT 15
  204 CONTINUE
      C(K) = CSAV(K)-DC(K)
      CALL PENFUN(IMINUS)
      IF(IFAIL.EQ.1) GO TO 502
      IF(IPRINT.NE.3) GO TO 205
      PRINT 4, IMINUS,(C(J),J=1,NC)
      PRINT 15
  205 CONTINUE
      D2(K)= (IPLUS-IMINUS)/(2.*DC(K))+AD(K)
      C(K) = CSAV(K)
      GO TO 51
   52 CONTINUE
      D2(K) = 0.
   51 CONTINUE
      GO TO 60
  502 DO 512 J=1,NC
      C(J) = CSAV(J)
```

```
  512 DC(J) = DC(J)/2.
      IFAIL = 0
      GO TO 50

C   GRADIENT CONVERGENCE TEST

   60 SUM = 0.
      DO 61 K=1,NC
   61 SUM = SUM+D2(K)**2
      G = SQRT(SUM)
      IF(G.GT..7*WT) GO TO 70

C   NOTE  -  VALUE USED IN GRADIENT CONVERGENCE TEST MAY BE MODIFIED
C          TO SUIT USER

      IF(IPRINT.NE.1) PRINT 62
      GO TO 300

C   UPDATE HH MATRIX

   70 CONTINUE

C   REDUCE D, D2, AND H VECTORS TO NCA   (ACTIVE DESIGN VARIABLES CNLY)

      NCA = NC
      IF(NC.EQ.1) GO TO 83
      K = NC-1
   79 CONTINUE
      IF(DC(K).NE.0.) GO TO 77
      NCA = NCA-1
      DO 78 KK=K,NCA
      D(KK) = D(KK+1)
      D2(KK) = D2(KK+1)
      H(KK) = H(KK+1)
      DIRCOS(KK) = DIRCOS(KK+1)
   78 CONTINUE
   77 CONTINUE
      K = K-1
      IF(K.EQ.0) GO TO 82
      GO TO 79
   82 CONTINUE
      IF(DC(NC).EQ.0.) NCA = NCA-1
   83 CONTINUE
```

```
      DO 71 K=1,NCA
      YT(K) = Y(K) = D2(K)-D(K)
   71 SIG(K) = S*DIRCOS(K)
      CALL MATRIX(MULT,1,NCA,NCA,YT,1,HH,16,YT,1)
      CALL MATRIX(MULT,NCA,1,NCA,Y,16,YT,1,B,16)
      CALL MATRIX(MULT,NCA,NCA,NCA,HH,16,B,16,B,16)
      SCLR = 0.
      DO 72 K=1,NCA
   72 SCLR = SCLR+YT(K)*Y(K)
      SCLR = 1./SCLR
      CALL MATRIX(ISCLR,NCA,NCA,0,SCLR,0,B,16,B,16)
      CALL MATRIX(ISUB,NCA,NCA,0,HH,16,B,16,HH,16)
      SCLR = 0.
      DO 73 K=1,NCA
      SIGT(K) = SIG(K)
   73 SCLR = SCLR+SIGT(K)*Y(K)
      SCLR = 1./SCLR
      CALL MATRIX(MULT,NCA,1,NCA,SIG,16,SIGT,1,A,16)
      CALL MATRIX(ISCLR,NCA,NCA,0,SCLR,0,A,16,A,16)
      CALL MATRIX(IADD,NCA,NCA,0,HH,16,A,16,HH,16)
      DO 74 K=1,NCA
   74 D(K) = D2(K)
      IF(KOUNTR.GE.(NCA+3)) GO TO 129
      IF (KOUNT.LT.NCA) GO TO 30

C     QUADRATIC CONVERGENCE TEST

      CALL MATRIX(MULT,1,NCA,NCA,D2,1,HH,16,D2,1)

C     EXPAND D AND D2 VECTORS TO NC

      IF(NC.EQ.1) GO TO 96
      K1 = NC-1
      DO 94 K=1,K1
      IF(DC(K).NE.0.) GO TO 94
      J1 = NCA
      DO 95 KK=K,NCA
      D(J1+1) = D(J1)
      D2(J1+1) = D2(J1)
      J1 = J1-1
   95 CONTINUE
      NCA = NCA+1
      D(K) = D2(K) = 0.
```

```
   94 CONTINUE
      IF(DC(NC).EQ.0.) D(NC) = D2(NC) = 0.
   96 CONTINUE
      Q = 0.
      DO 75 K=1,NC
   75 Q = Q+C2(K)*D(K)
      Q = 1./2.*Q/(WT+PF)
      IF(IPRINT.NE.1) PRINT 998, Q
  998 FORMAT(/* Q =*E16.8)
      IF(Q.GT.0.) GO TO 80
      IF(IPRINT.NE.1) PRINT 76
      GO TO 20
   80 IF(Q.GT..02) GO TO 90
      IF(IPRINT.NE.1) PRINT 81
      GO TO 300

C    MINIMUM MOVE SIZE CONVERGENCE TEST

   90 SUM = 0.
      DO 91 K=1,NC
   91 SUM = SUM+D(K)**2
      SUM = SQRT(SUM)
      DCR = 0.
      DO 92 K=1,NC
      DIRCOS(K) = -D(K)/SUM
      IF(ABS(DIRCOS(K)).LT.DCR) GO TO 92
      KK = K
      DCR = ABS(DIRCOS(K))
   92 CONTINUE
      DO 93 K=1,NC
      H(K) =  20.*DIRCOS(K)*DC(KK)
      C(K) = CSAV(K)+H(K)
   93 CONTINUE
      KDERIV = 0
      CALL PENFUN(M2)
      IF(IFAIL.EQ.0) GO TO 150
  100 IF(IPRINT.NE.1) PRINT 101
      IFAIL = 0
      GO TO 300
  150 CONTINUE
      IF(IPRINT.NE.3) GO TO 206
      PRINT 6, M2,(C(J),J=1,NC)
    6 FORMAT(/* M2 =*E16.8/(5E25.8))
```

```
      PRINT 15
  206 CONTINUE
      DO 151 K=1,NC
  151 C(K) = CSAV(K)+1.5*H(K)
      KDERIV = 1
      CALL PENFUN(MPLUS)
      IF(IFAIL.EQ.0) GO TO 120
      GO TO 100
  120 SUM = 0.
      IF(IPRINT.NE.3) GO TO 207
      PRINT 7, MPLUS,(C(J),J=1,NC)
    7 FORMAT(/* MPLUS =*E16.8/(5E25.8))
      PRINT 15
  207 CONTINUE
      DO 121 K=1,NC
  121 SUM = SUM+H(K)**2
      DM2 = (MPLUS-M2)/(.5*SQRT(SUM))
      DO 122 K=1,NC
  122 DM2 = DM2+DIRCOS(K)*AD(K)
      IF(DM2.LT.0.) GO TO 49
      GO TO 100
  129 IF(IPRINT.NE.1) PRINT 128, KOUNTR,R
  128 FORMAT(/* NCA+3 =*I3* ONEDIMENSIONAL SEARCHES HAVE BEEN CARRIED OU
     1T FOR R =*E16.8*  - CONVERGENCE ASSUMED*/)
  300 CTEST = PF/(WT+PF)
      IF(IPRINT.NE.1) PRINT 921, CTEST
      IF(CTEST.LT.0.) GO TO 20
      IF(CTEST.GT..02) GO TO 310
      PRINT 15
      PRINT 306
  306 FORMAT(* THE FOLLOWING INFORMATION IS FOR THE FINAL DESIGN*)
      IF(IPRINT.EQ.1) PRINT 921, CTEST
      PRINT 301
      DO 305 J=1,NC
  305 C(J)=CSAV(J)
      CALL WEIGHT(FWT)
      PRINT 309, FWT
  309 FORMAT(//* FINAL WEIGHT=*E16.8/)
      PRINT 308, (C(J),J=1,NC)
  308 FORMAT(//* FINAL CONSTANTS*/(4E20.8))
      NEWR = 1
      CALL PENFUN(I)
      CALL TABLE
```

```
      RETURN                                          88000000
  310 R = R/RDC                                       88100000
      IF(IPRINT.NE.1) PRINT 311, R                    88200000
      PF = PF/RDC                                     88300000
      IF(IPRINT.NE.1) PRINT 1, PF                     88400000
      IF(IPRINT.NE.3) GO TO 10                        88500000
      PRINT 15                                        88600000
      PRINT 5                                         88700000
      GO TO 10                                        88800000
      END                                             88900000
```

```
      SUBROUTINE ONEDIM

C     ONE-DIMENSIONAL SEARCH USING DIRECTION COSINES CALCULATED IN FPOWEL

  332 FORMAT(/* S =*E16.8/
     1* B*E16.8/* D2,D3*2E16.8,5X*E2,E3*2E16.8,5X*F2,F3*2E16.8/
     1* P =*E16.8,5X* PPRIME =*E16.8)
  334 FORMAT(/* G1,G2,G3*3E16.8)
  335 FORMAT(* S(*I2*) =*E16.8)
  338 FORMAT(///* J =*I3)
  800 FORMAT(/* VO,VP,VPP,*3E20.8/* OMEGAO,OMEGAP,OMEGAPP*3E20.8/
     1* V1,OMEGA1*2E20.8)

      EXTERNAL FOFS
      LOGICAL DESCON
      REAL MU
      DIMENSION H(16)
      DIMENSION A(3,5)
      DIMENSION S(10),G1(10),G2(10),G3(10)
      DIMENSION SA(10)
      DIMENSION OMA(10),VA(10)
      COMMON
     2/BLK2/OMEGA,V
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     5/BLK5/OMEGA1,DELOM,OMFIN,V1,DELV,VFIN
     1/BLK6/IOPT,IPRINT,ISNDWCH
     2/BLK12/VCR,R,NC,H,DESCON(3)
     7/BLK17/NEWR
     2/BLK20/IFIRST,IFAIL,NOCONV,NOS
     1/BLK28/CSAV(16),DC(16),D(16),DIRCOS(16),DM1,PF
     3/BLK30/B,D2,D3,E2,E3,F2,F3,LINEAR,VP,G,VPP
     1/BLK31/DCR,OMPLUS,VPLUS
     2/BLK32/IG
     3/BLK33/GG1,GG2,GG3
     4/BLK34/GP1,GP2,GP3
     5/BLK35/OMEGAO,VO
     7/BLK37/G11,G22,G33
     4/BLK40/AD(16)
     1/BLK41/WT
     3/BLK43/SRETURN
      IF(DESCON(1)) 200,201
  201 CONTINUE
```

```
      VPLUS = OMPLUS = 2.
      V = VO = OMEGA = OMEGAO = 2.
      VPP = OMEGAPP = O.
      VCR = 1.
      G1APRX = 1.
  200 CONTINUE
      VP = (VPLUS-VO)/DCR
      OMEGAP = (OMPLUS-OMEGAO)/DCR
      OMA(1) = OMEGAO
      VA(1) = VO
      EP1 = 1.E-3
      EP2 = 1.E-4
      MAXI = 100
      J = 1
      IF(IPRINT.EQ.3) PRINT 338, J
      S(1) = O.
      SA(1) = O.

C     LINEAR APPROXIMATION TO G AND F USED FOR PPRIME

      AP = O.
      B = O.
      DO 100 I=1,NC
      AP = AP+CSAV(I)*AD(I)
      B = B+AD(I)*DIRCOS(I)
  100 CONTINUE
      D2 = G2(1) = 1.-G22
      D3 = G3(1) = 1.-G33
      E2 = (G22-GP2) /DCR
      E3 = (G33-GP3) /DCR
      F1 = O.
      V1 = VO+S(1)*VP
      G1(1) = 1.-VCR/V1
      PPRIME = B-R*( VCR*VP/(V1**2*G1(1)**2)+30.*E2/(D2+E2*S(1))**2
     1+30.*E3/(D3+E3*S(1))**2)
      P = AP+B*S(1)+R*(1./G1(1)+30./(D2+E2*S(1))+30./(D3+E3*S(1)))
      IF(IPRINT.EQ.3) PRINT 332, S(1),B,D2,D3,E2,E3,F2,F3,P,PPRIME
      IF(PPRIME.LE.O.) GO TO 110
      PRINT 10
   10 FORMAT(//* PPRIME POSITIVE AT S = O*//)
      STOP
  110 CONTINUE
```

```
C     LOCATE S(2) USING LINEAR APPROXIMATION

      J = 2
      SI = S(1)
      SF = .2
      SDEL = 20.*DCR
      LINEAR = 1
      G = G1(1)
      CALL ITR2(S(2),SI,SF,SDEL,FOFS,EP1,EP2,MAXI,ICODE)
      IF(ICODE.EQ.0) GO TO 5
      IF(ICODE.NE.3) GO TO 1000
      S(2) = .08
      GO TO 76
    5 S(2) = .4*S(2)
   76 CONTINUE
      IF(IPRINT.EQ.3) PRINT 338, J
   77 CONTINUE
      V1 = VC+S(2)*VP
      OMEGA1 = OMEGA0+S(2)*OMEGAP
    1 CONTINUE
      DO 78 I=1,NC
      C(I) = CSAV(I)+S(2)*DIRCOS(I)
   78 CONTINUE
      IF(DESCON(1)) 202,2
  202 CONTINUE
      IG = 1
      CALL PENFUN(X)
      IF(IFAIL.EQ.0) GO TO 79
      IFAIL = 0
      S(2) = .5*S(2)
      GO TO 77
   79 CONTINUE
      VA(2) = V
      OMA(2) = OMEGA
      SA(2) = S(2)
      VPP = (V-VO-S(2)*VP)/S(2)**2
      OMEGAPP = (OMEGA-OMEGA0-S(2)*OMEGAP)/S(2)**2
   80 CONTINUE
      V1 = V0+S(2)*VP+S(2)**2*VPP
      G1APRX = 1.-VCR/V1
      IF(G1APRX.GT.0.) GO TO 2
      S(2) = .8*S(2)
      V1 = V0+S(2)*VP+S(2)**2*VPP
```

```
      OMEGA1 = OMEGA0+S(2)*OMEGAP+S(2)**2*OMEGAPP
      GO TO 1
    2 CONTINUE
      IG = 2
      IF(IPRINT.EQ.3) PRINT 800, V0,VP,VPP,OMEGA0,OMEGAP,OMEGAPP,V1,
     1OMEGA1
      CALL PENFUN(X)
      IF(IFAIL.EQ.0) GO TO 6
      IFAIL = 0
      S(2) = .5*S(2)
      DO 81 I=1,NC
      C(I) = CSAV(I)+S(2)*DIRCOS(I)
   81 CONTINUE
      IF(DESCCN(1)) 203,2
  203 CONTINUE
      GO TO 80
    6 CONTINUE
      G1(2) = G1APRX
      G2(2) = 1.-GG2
      G3(2) = 1.-GG3

C   CALCULATE PARABOLIC APPROXIMATION TO G2 AND G3 USING VALUES AT S(2)

      F2 = (G2(2)-D2-E2*S(2))/S(2)**2
      F3 = (G3(2)-D3-E3*S(2))/S(2)**2
    3 CONTINUE
      PPRIME = B-R*(VCR*(VP+2.*S(2)*VPP)/(V0+S(2)*VP+S(2)**2*VPP-VCR)**2
     1+(E2+2.*F2*S(2))/(D2+E2*S(2)+F2*S(2)**2)**2*30.
     2+(E3+2.*F3*S(2))/(D3+E3*S(2)+F3*S(2)**2)**2*30.)
      P = AP+B*S(2)+R*(1./G1(2)
     1+30./(D2+E2*S(2)+F2*S(2)**2)+30./(D3+E3*S(2)+F3*S(2)**2))
      IF(IPRINT.EQ.3) PRINT 332, S(2),B,D2,D3,E2,E3,F2,F3,P,PPRIME
    4 CONTINUE

C   LOCATE S(3) USING PARABOLIC APPROXIMATION

      J = 3
      SI = S(2)
      SF = S(2)+PPRIME/(-ABS(PPRIME))*2.*S(2)
      IF(SI.LT.SF) GO TO 15
      X = SI
      SI = SF
      SF = X
```

```
 15 CONTINUE
    SDEL = S(2)/10.
    LINEAR = 0
    G = G1(2)
    CALL ITR2(S(3),SI,SF,SDEL,FOFS,EP1,EP2,MAXI,ICODE)
    IF(ICODE.EQ.0) GO TO 18
    IF(ICOCE.NE.3) GO TO 1000
    S(3) = SF
    IF(PPRIME.GT.0.) S(3) = SI
    GO TO 11
 18 S(3) = S(2)+.6*(S(3)-S(2))
 11 CONTINUE
    IF(IPRINT.EQ.3) PRINT 338, J
    DO 88 I=1,NC
    C(I) = CSAV(I)+S(3)*DIRCOS(I)
 88 CONTINUE
    IF(DESCCN(1)) 212,12
212 CONTINUE
    V1 = VO+S(3)*VP+S(3)**2*VPP
    OMEGA1 = OMEGAO+S(3)*OMEGAP+S(3)**2*OMEGAPP
    IG = 1
    CALL PENFUN(X)
    IF(IFAIL.EQ.0) GO TO 89
    IFAIL = 0
    S(3) = S(2)+.8*(S(3)-S(2))
    GO TO 11
 89 CONTINUE
    SA(3) = S(3)
    VA(3) = V
    OMA(3) = OMEGA
    VPP = ((VA(2)-VO)/SA(2)-(VA(3)-VO)/SA(3))/(SA(2)-SA(3))
    VP = (VA(3)-VO)/SA(3)-VPP*SA(3)
    OMEGAPP = ((OMA(2)-OMEGAO)/SA(2)-(OMA(3)-OMEGAO)/SA(3))/(SA(2)-SA(
   13))
    OMEGAP = (OMA(3)-OMEGAO)/SA(3)-CMEGAPP*SA(3)
 90 CONTINUE
    V1 = VO+S(3)*VP+S(3)**2*VPP
    G1APRX = 1.-VCR/V1
    IF(G1APRX.GT.0.) GO TO 12
    S(3) = S(2)+.8*(S(3)-S(2))
    GO TO 11
 12 CONTINUE
    IG = 2
```

```
      CALL PENFUN(X)
      IF(IFAIL.EQ.0) GO TO 121
      IFAIL = 0
      S(3) = S(2)+.6*(S(3)-S(2))
      DO 91 I=1,NC
      C(I) = CSAV(I)+S(3)*DIRCOS(I)
   91 CONTINUE
      IF(DESCON(1)) 213,12
  213 CONTINUE
      GO TO 90
  121 CONTINUE
      G1(3) = G1APRX
      G2(3) = 1.-GG2
      G3(3) = 1.-GG3

C     CALCULATE PARABOLIC APPROXIMATION TO G2 AND G3 USING VALUES AT S(3)

      F2 = ((G2(3)-D2)/S(3)-(G2(2)-D2)/S(2))/(S(3)-S(2))
      E2 = (G2(3)-D2)/S(3)-F2*S(3)
      F3 = ((G3(3)-D3)/S(3)-(G3(2)-D3)/S(2))/(S(3)-S(2))
      E3 = (G3(3)-D3)/S(3)-F3*S(3)
   13 CONTINUE
      PPRIME = B-R*(VCR*(VP+2.*S(3)*VPP)/(VO+S(3)*VP+S(3)**2*VPP-VCR)**2
     1+(E2+2.*F2*S(3))/(D2+E2*S(3)+F2*S(3)**2)**2*30.
     2+(E3+2.*F3*S(3))/(D3+E3*S(3)+F3*S(3)**2)**2*30.)
      P = AP+B*S(3)+R*(1./G1(3)
     1+30./(D2+E2*S(3)+F2*S(3)**2)+30./(D3+E3*S(3)+F3*S(3)**2))
      IF(IPRINT.EQ.3) PRINT 332, S(3),B,D2,D3,E2,E3,F2,F3,P,PPRIME
   14 CONTINUE

C     LOCATE S(4) USING PARABOLIC APPROXIMATION

      SI = S(3)
      SF = S(3)+PPRIME/(-ABS(PPRIME))*30.*ABS(S(3)-S(2))
      IF(SI.LT.SF) GO TO 16
      X = SI
      SI = SF
      SF = X
   16 CONTINUE
      SDEL = ABS(S(3)-S(2))/10.
      G = G1(3)
      CALL ITR2(S(4),SI,SF,SDEL,FOFS,EP1,EP2,MAXI,ICODE)
      IF(ICODE.EQ.0) GO TO 20
```

```
      IF(ICODE.NE.3) GO TO 1000
      S(4) = SF
      IF(PPRIME.GT.0.) S(4) = SI
   20 CONTINUE
      J = 4
   21 CONTINUE
      IF(IPRINT.EQ.3) PRINT 338, J
      DO 98 I=1,NC
      C(I) = CSAV(I)+S(J)*DIRCCS(I)
   98 CONTINUE
      IF(DESCON(1)) 222,22
  222 CONTINUE
      V1 = VC+S(J)*VP+S(J)**2*VPP
      OMEGA1 = OMEGAO+S(J)*OMEGAP+S(J)**2*OMEGAPP
      IF(IPRINT.EQ.3) PRINT 800, VO,VP,VPP,OMEGAO,OMEGAP,OMEGAPP,V1,
     1OMEGA1
      IG = 1
      CALL PENFUN(X)
      IF(IFAIL.EQ.0) GO TO 99
      IFAIL = 0
      S(J) = S(J-1)+.8*(S(J)-S(J-1))
      GO TO 21
   99 CONTINUE
      SA(J) = S(J)
      VA(J) = V
      OMA(J) = OMEGA
      KOUNT = 0

C   CALCULATE PARABOLIC APPROXIMATION TO V AND OMEGA USING SA(J) AND 2 CLOSEST
C       VALUES OF SA

      IF(IPRINT.NE.3) GO TO 50
      PRINT 330
  330 FORMAT(/* S FOR PARABOLIC APPROXIMATION TO FLUTTER CONSTRAINT*)
      PRINT 335, (I,SA(I),I=1,J)
   50 CONTINUE
      SAVE = SMINI1 = ABS(SA(J)-SA(1))
      SMINI2 = ABS(SA(J)-SA(2))
      IND1 = 1
      IND2 = 2
      IF(SMINI2.GE.SMINI1) GO TO 620
      SMINI1 = SMINI2
      SMINI2 = SAVE
```

```
      IND1 = 2
      IND2 = 1
  620 CONTINUE
      J1 = J-1
      DO 660 I=3,J1
      SAVE = ABS(SA(J)-SA(I))
      IF(SAVE.GT.SMINI2) GO TO 660
      SMINI2 = SAVE
      IND2 = I
      IF(SMINI1.LT.SMINI2) GO TO 660
      SAVE = SMINI2
      SMINI2 = SMINI1
      SMINI1 = SAVE
      IND2 = IND1
      IND1 = I
  660 CONTINUE
      A(1,1) = A(2,1) = A(3,1) = 1.
      A(1,2) = SA(J)
      A(1,3) = SA(J)**2
      A(2,2) = SA(IND1)
      A(2,3) = SA(IND1)**2
      A(3,2) = SA(IND2)
      A(3,3) = SA(IND2)**2
      A(1,4) = VA(J)
      A(2,4) = VA(IND1)
      A(3,4) = VA(IND2)
      A(1,5) = OMA(J)
      A(2,5) = OMA(IND1)
      A(3,5) = OMA(IND2)
      ISOL = 10
      CALL MATRIX(ISOL,3,5,0,A,3,DET)
      VO = A(1,4)
      VP = A(2,4)
      VPP = A(3,4)
      OMEGAO = A(1,5)
      OMEGAP = A(2,5)
      OMEGAPP = A(3,5)
  101 CONTINUE
      V1 = VO+S(J)*VP+S(J)**2*VPP
      G1APRX = 1.-VCR/V1
      IF(G1APRX.GT.0.) GO TO 22
      S(J) = S(J-1)+.8*(S(J)-S(J-1))
      GO TO 21
```

```
   22 CONTINUE
      IG = 2
      CALL PENFUN(X)
      IF(IFAIL.EQ.0) GO TO 122
      IFAIL = 0
      S(J) = S(J-1)+.8*(S(J)-S(J-1))
      DO 102 I=1,NC
      C(I) = CSAV(I)+S(J)*DIRCOS(I)
  102 CONTINUE
      KOUNT = 1
      IF(DESCON(1)) 223,22
  223 CONTINUE
      GO TO 101
  122 CONTINUE
      G1(J) = G1APRX
      G2(J) = 1.-GG2
      G3(J) = 1.-GG3

C   CALCULATE PARABOLIC APPROXIMATION TO G2 AND G3 USING S(J) AND TWO CLCSEST
C      VALUES OF S

C     IF(KOUNT.EQ.0.AND.DESCON(1))  GO TO 651
      IF(IPRINT.NE.3) GO TO 51
      PRINT 331
  331 FORMAT(//* S FOR PARABOLIC APPROXIMATION TO STRESS AND MINGAGE CON
     1STRAINTS*)
      PRINT 335, (I,S(I),I=1,J)
   51 CONTINUE
      SAVE = SMINI1 = ABS(S(J)-S(1))
      SMINI2 = ABS(S(J)-S(2))
      IND1 = 1
      IND2 = 2
      IF(SMINI2.GE.SMINI1) GO TO 610
      SMINI1 = SMINI2
      SMINI2 = SAVE
      IND1 = 2
      IND2 = 1
  610 CONTINUE
      J1 = J-1
      DO 650 I=3,J1
      SAVE = ABS(S(J)-S(I))
      IF(SAVE.GT.SMINI2) GO TO 650
      SMINI2 = SAVE
```

```
      IND2 = I
      IF(SMINI1.LT.SMINI2) GO TC 650
      SAVE = SMINI2
      SMINI2 = SMINI1
      SMINI1 = SAVE
      IND2 = IND1
      IND1 = I
  650 CONTINUE
  651 CONTINUE
      A(1,1) = A(2,1) = A(3,1) = 1.
      A(1,2) = S(J)
      A(1,3) = S(J)**2
      A(2,2) = S(IND1)
      A(2,3) = S(IND1)**2
      A(3,2) = S(IND2)
      A(3,3) = S(IND2)**2
      A(1,4) = G2(J)
      A(2,4) = G2(IND1)
      A(3,4) = G2(IND2)
      A(1,5) = G3(J)
      A(2,5) = G3(IND1)
      A(3,5) = G3(IND2)
      ISOL = 10
      CALL MATRIX(ISOL,3,5,0,A,3,DET)
      D2 = A(1,4)
      E2 = A(2,4)
      F2 = A(3,4)
      D3 = A(1,5)
      E3 = A(2,5)
      F3 = A(3,5)
   23 CONTINUE
      PPRIME = B-R*(VCR*(VP+2.*S(J)*VPP)/(VO+S(J)*VP+S(J)**2*VPP-VCR)**2
     1+(E2+2.*F2*S(J))/(D2+E2*S(J)+F2*S(J)**2)**2*30.
     2+(E3+2.*F3*S(J))/(D3+E3*S(J)+F3*S(J)**2)**2*30.)
      P = AP+B*S(J)+R*(1./G1(J)
     1+30./(D2+E2*S(J)+F2*S(J)**2)+30./(D3+E3*S(J)+F3*S(J)**2))
      IF(IPRINT.EQ.3) PRINT 332, S(J),B,D2,D3,E2,E3,F2,F3,P,PPRIME
      IF(ABS(PPRIME).GT..2*ABS(CM1)) GC TO 24

C     SLOPE CONVERGENCE TEST AT S(J)

      IF(KOUNT.EQ.0) GO TO 527
      IF(DESCON(1)) 227,527
```

```
227 CONTINUE
    KOUNT = 0
520 CONTINUE
    V1 = V0+S(J)*VP+S(J)**2*VPP
    OMEGA1 = CMEGA0+S(J)*OMEGAP+S(J)**2*CMEGAPP
    IG = 1
    CALL PENFUN(X)
    G1(J) = 1.-GG1
    IF(IPRINT.EQ.3) PRINT 334, G1(J),G2(J),G3(J)
    IF(IFAIL.EQ.0) GO TO 521
    IFAIL = 0
    S(J) = S(J-1)+.8*(S(J)-S(J-1))
    DO 301 I=1,NC
    C(I) = CSAV(I)+S(J)*DIRCOS(I)
301 CONTINUE
    KOUNT = 1
    GO TO 520
521 CONTINUE
    IF(KOUNT)525,525,522


522 CONTINUE
    G1APRX = G1(J)
    GO TO 23
525 CONTINUE
    IF(ABS(G1(J)-G1APRX).GT..02) GO TO 526
527 CONTINUE
    IG = 0
    NEWR = 1
    IF(IPRINT.EQ.1) NEWR=0
    CALL WEIGHT(WT)
    OMEGA1 = OMEGAO + S(J) * CMEGAP + S(J)**2 * OMEGAPP
    V1 = V0 + S(J) * VP + S(J)**2 * VPP
    CALL PENFUN(PF)
    IF(IFAIL.EQ.0) GO TO 524
    IFAIL = 0
    PRINT 340
340 FORMAT(/* A FAILURE OCCURRED AFTER THE SLOPE CONVERGENCE TEST IN S
   1UBROUTINE ONEDIM WAS SATISFIED.*/* CONSTRAINTS DO NOT APPEAR TO BE
   2 UNIMODAL.*/)
    GO TO 21
524 CONTINUE
    OMEGA1=OMEGA0=OMEGA
```

```
      V1=V0=V
      SRETURN = S(J)
      G11 = GG1
      G22 = GG2
      G33 = GG3
      NEWR = 0
      DO 523 I=1,NC
      CSAV(I) = CSAV(I)+S(J)*DIRCOS(I)
  523 CONTINUE
      RETURN
  526 CONTINUE
      IF(KOUNT.EQ.0) GO TO 24
      PPRIME = B-R*(VCR*(VP+2.*S(J)*VPP)/(V0+S(J)*VP+S(J)**2*VPP-VCR)**2
     1+(E2+2.*F2*S(J))/(D2+E2*S(J)+F2*S(J)**2)**2*30.
     2+(E3+2.*F3*S(J))/(D3+E3*S(J)+F3*S(J)**2)**2*30.)
      P = AP+B*S(J)+R*(1./G1(J)
     1+30./(D2+E2*S(J)+F2*S(J)**2)+30./(D3+E3*S(J)+F3*S(J)**2))
      IF(IPRINT.EQ.3) PRINT 332, S(J),B,D2,D3,E2,E3,F2,F3,P,PPRIME
   24 CONTINUE

C     LOCATE S(J+1) USING PARABOLIC APPROXIMATION

      SI = S(J)
      SF = S(J)+PPRIME/(-ABS(PPRIME))*30.*ABS(S(J)-S(J-1))
      IF(SI.LT.SF) GO TO 17
      X = SI
      SI = SF
      SF = X
   17 CONTINUE
      SDEL = ABS(S(J)-S(J-1))/10.
      G = G1(J)
      CALL ITR2(S(J+1),SI,SF,SDEL,FOFS,EP1,EP2,MAXI,ICODE)

      IF(ICODE.EQ.0) GO TO 31
      IF(ICODE.NE.3) GO TO 1000
      S(J+1) = SF
      IF(PPRIME.GT.0.) S(J+1) = SI
   31 CONTINUE
      J = J+1
      IF(J.LT.11) GO TO 21
      PRINT 1002
 1002 FORMAT(/* ONEDIM    J=10    STOP*)
      STOP
```

```
1000 PRINT 1001, ICODE                                                        140500000
1001 FORMAT(//* ERROR RETURN FRCM ITR2    ICODE =*I2)                         140600000
     STOP                                                                     140700000
     END                                                                      140800000
```

```
      FUNCTICN FOFS(S)
C   CALLED BY ITR2 IN ONEDIM TO APPRCXIMATE P FUNCTION

      LOGICAL DESCON
      DIMENSION H(16)
      COMMON
     2/BLK12/VCR,R,NC,H,DESCON(3)
     3/BLK30/B,D2,D3,E2,E3,F2,F3,L,VP,G,VPP
     5/BLK35/OMEGAO,VO
      IF(L.EQ.1) GO TO 100
      FOFS = B-R*(VCR*(VP+2.*S*VPP)/(VO+S*VP+S*S*VPP-VCR)**2
     1+30.*(E2+2.*F2*S)/(D2+E2*S+F2*S**2)**2
     2+30.*(E3+2.*F3*S)/(D3+E3*S+F3*S**2)**2)
      RETURN
  100 CONTINUE
      FOFS = B-R*(VCR*VP/(VO+S*VP-VCR)**2
     1                           +30.*E2/(D2+E2*S)**2+30.*E3/(D3+E3
     1*S)**2)
      RETURN
      END
```

```
      SUBROUTINE WEIGHT(WT)                                             143000000
      REAL MU                                                           143100000
      COMMON                                                            143200000
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS                                  143300000
     3/BLK40/AD(16)                                                     143400000
     1/BLK12/VCR,R,NC                                                   143500000
      WT = 0.                                                           143600000
      DO 100 I=1,NC                                                     143700000
  100 WT = WT+C(I)*AD(I)                                                143800000
      RETURN                                                            143900000
      END                                                               144000000
```

```
      SUBROUTINE PENFUN(PF)

C    CALCULATES PENALTY FUNCTION

      LOGICAL DESCON
      REAL MU
      DIMENSION H(16)
      COMMON
     2/BLK2/CMEGA,V
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     8/BLK8/KDERIV
     1/BLK12/VCR,R,NC,H,DESCON(3)
     7/BLK17/NEWR
     1/BLK20/IFIRST,IFAIL,NOCONV,NOS
     2/BLK32/IG
     3/BLK33/G1,G2,G3

C    BEGIN FLUTTER CONSTRAINT.  FLUTTER VELOCITY V MUST BE GREATER THAN VCR,
C    THE CRITICAL FLUTTER VELOCITY.

      IF(IG.EQ.2) GO TO 60
      IF(DESCCN(1)) GO TO 50
      G1 = 0.
      GO TO 60
   50 CONTINUE
      NOCONV = 0
      CALL FLUTTER
      IF(IFIRST.EQ.1.AND.V.LT.VCR) PRINT 666
  666 FORMAT(/* V .LT. VCR  -  STOP*)
      IF(IFIRST.EQ.1.AND.V.LT.VCR) STOP
      IF(IPRINT.EQ.1) GO TO 10
      IF(IFIRST.EQ.2.AND.NOCONV.EQ.1) PRINT 667
  667 FORMAT(/* NOCONV - RETURN*)
   10 IF(IFIRST.EQ.2.AND.NOCCNV.EQ.1) GO TO 100
      IF(IPRINT.EC.1) GO TO 20
      IF(IFIRST.EQ.2.AND.V.LT.VCR) PRINT 668
  668 FORMAT(/* V.LT.VCR - REDUCE S OR DC*)
   20 IF(IFIRST.EQ.2.AND.V.LT.VCR) GC TO 100
      GO TO 400
  100 CONTINUE
      IFAIL=1
```

```
      RETURN
  400 CONTINUE
      G1 = VCR/V

C   BEGIN STRESS CONSTRAINT.  STRESS CONDITION Y AT EACH POINT ON THE WING
C   SURFACE MUST BE LESS THAN YCR, THE CRITICAL STRESS CONDITION.

   60 CONTINUE
      IF(IG.EQ.1) GO TO 777
      IF(DESCON(2)) GO TO 61
      G2 = 0.
      GO TO 70
   61 CONTINUE
      IFAIL=0
      CALL STRESS(G2)
      IF(IFAIL.EQ.1) RETURN

C   BEGIN MINIMUM GAGE CONSTRAINT.  TBAR IS THE DEPTH OF THE WING IF THE WING
C   IS SOLID.  TBAR IS THE THICKNESS OF THE UPPER OR LOWER FACE IF THE WING
C   IS A SANDWICH.  TBAR MUST BE GREATER THAN TBARMIN, THE MINIMUM THICKNESS
C   ALLOWED.

   70 IF(DESCON(3)) GO TO 80
      G3 = 0.
      GO TO 777
   80 CONTINUE
      CALL MINGAGE(G3)
      IF(IFAIL.EQ.1) RETURN
  777 CONTINUE

C   IN THIS PROGRAM G1, G2, AND G3 ARE THE RATIOS OF THE BEHAVIOR
C   VARIABLES FLUTTER, STRESS, AND MINIMUM GAGE TO THE CRITICAL VALUES
C   OF THESE VARIABLES.   ALL COMPUTATIONS CARRIED OUT IN THE PROGRAM
C   USE THAT DEFINITION.  FOR EXAMPLE, G1 = VCR/V.  THE QUANTITIES
C   CALLED G1, G2, G3 IN THE OUTPUT AGREE WITH CONVENTIONAL NOTATION.
C   FOR EXAMPLE, G1(OUTPUT) = 1-G1(INTERNAL).
C   ALSO, FAILURE RATIO 1(OUTPUT) = G1(INTERNAL).

      GSUB1=1.-G1
      GSUB2=1.-G2
      GSUB3=1.-G3
      IF(IPRINT.EQ.3.OR.NEWR.EQ.1) PRINT 1, G1,GSUB1,G2,GSUB2,G3,GSUB3
    1 FORMAT(/* FAILURE RATIO 1 =*E16.8,5X*G1 =*E16.8/* FAILURE RATIO 2
```

```
1=*E16.8,5X*G2 =*E16.8/* FAILURE RATIC 3 =*E16.8,5X*G3 =*E16.8)
 SUM = 1./(1.-G1)+1./(1.-G2)*30.+1./(1.-G3)*30.
 PF = R*SUM
 NEWR = C
 RETURN
 END
```

```
      SUBROUTINE STRESS(G2)

C    THIS SUBROUTINE SOLVES FOR DEFLECTIONS CAUSED BY A STATIC LOAD ON THE
C    SURFACE OF THE WING.  THE DEFLECTION IS IN TERMS OF PHI.  THE LOADING
C     IS A UNIFORM PRESSURE LOAD DENOTED P OF INTENSITY PLOAD (PSI).

      LOGICAL DESCON
      EXTERNAL FUNCS
      EXTERNAL FSTRESS
      REAL MU
      COMPLEX AAC
      DIMENSION H(16)
      DIMENSION EPS4P(33,1),P(11,1),
     1AST(33,33),IPIVOT(33),
     1SX(11),FOFX(1),ANS(1)
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     1/BLK11/EPS4P
     1/BLK20/IFIRST,IFAIL,NOCONV,NOS
     1/BLK13/Y,C1,C2
     2/BLK14/YCR,IY
     1/BLK15/PLOAD
     7/BLK17/NEWR
     1/BLK12/VCR,R,NC,H,DESCON(3)
     2/BLK32/IG
     1/NEWBLK/AAC(33,33),ASTRESS(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLKNPS/AREA
      NAMELIST/STRSS/PLOAD,YCR
      NMAX=33
      ISCLR = 24

C    LOADING IS A COLUMN VECTOR CALLED P

      IF(IPRINT.EQ.0) GO TO 151
      IF(IG.EQ.2) GO TO 152
      IF(DESCON(1)) GO TO 151
  152 CONTINUE
      CALL AAMAT
      CALL SMAT
  151 CONTINUE
```

```
      IF(IPRINT.NE.0) GO TO 150
      READ STRSS
      PRINT 10
      PRINT 11, ((ASTRESS(I,J),J=1,NN1),I=1,NN33)
      PRINT 13
      PRINT 11, ((ASTRESS(I,J),J=NN2,NN22),I=1,NN33)
      PRINT 13
      PRINT 11, ((ASTRESS(I,J),J=NN23,NN33),I=1,NN33)
   10 FORMAT(1H1///* ASTRESS MATRIX  (ORIGINAL AA MATRIX USED TO BUILD S
     1TRESS MATRIX)*//)
   11 FORMAT(5E16.8)
   13 FORMAT(1H1//////)
  150 CONTINUE

C   CALCULATE P BY ANALYTIC INTEGRATION, MULTIPLY BY EPS**4,
C   AND STORE IN EPS4P

      EPS4=EPS**4
      DO 200 I=1,NN1
      Y=EPS*FLOAT(I)
      CALL CEE
      P(I,1)=PLOAD*(C2-C1)
      IF(I.EQ.NN1-1) P(I,1)=P(I,1)/2.
      IF(I.EQ.NN1) P(I,1)=0.
      EPS4P(I,1)=EPS4*P(I,1)
      II=I+NN1
      P(I,1)=PLOAD*(C2**2/2.-C1**2/2.)
      IF(I.EQ.NN1-1) P(I,1)=P(I,1)/2.
      IF(I.EQ.NN1) P(I,1)=0.
      EPS4P(II,1)=EPS4*P(I,1)
      III=I+NN22
      P(I,1)=PLOAD*(C2**3/3.-C1**3/3.)
      IF(I.EQ.NN1-1) P(I,1)=P(I,1)/2.
      IF(I.EQ.NN1) P(I,1)=0.
  200 EPS4P(III,1)=EPS4*P(I,1)
      IORD=NN33
      DO 600 I=1,IORD
      DO 600 J=1,IORD
  600 AST(I,J) = ASTRESS(I,J)
      IF(IPRINT.NE.0) GO TO 55
      PRINT 15, PLOAD
   15 FORMAT(///* PLOAD =*E16.8//)
      PRINT 16, (EPS4P(I,1),I=1,IORD)
```

```
   16 FORMAT(///6X*EPS4P*/(E16.8))
   55 CONTINUE

C     SOLVE FOR PHI AND STORE IN EPS4P

      CALL SIMEQ(AST,IORD,EPS4P,1,DETERM,IPIVOT,NMAX,ISCALE)
      IF(IPRINT.NE.0) GO TO 250
      PRINT 251, (EPS4P(I,1),I=1,IORD)
  251 FORMAT(///7X*PHI*/(E16.8))
      RETURN
  250 CONTINUE
      DO 500 I=1,NN1
      IY = I
      Y = EPS*FLOAT(I-1)
      IF(NEWR.EQ.1.AND.I.EQ.1) PRINT 651
  651 FORMAT(//* STRESS TABLE*/)
      IF(NEWR.EQ.1) PRINT 650, Y
  650 FORMAT(/* Y =*E16.8/4X*X*,13X*-ZU*,11X*EP X*,10X*EP Y*,10X*GAM XY*
     18X*SIG X*,9X*SIG Y*,9X*TAU XY*,8X*Y*)
      CALL CEE
      CALL MGAUSS(C1,C2,NCS,ANS,FSTRESS,FOFX,1)
      IF(IFAIL.EQ.1) RETURN
  500 SX(I) = ANS(1)
      SUM = 1./2.*SX(1)+1./2.*SX(NN1)
      DO 300 I=2,NN
  300 SUM = SUM+SX(I)
      SUM = SUM*EPS/AREA
      G2 = 1.-1./SUM
      RETURN
      END
```

```
      SUBROUTINE FSTRESS(X,FOFX)

C     THIS SUBROUTINE TAKES THE STRAINS CALCULATED IN SUBROUTINE EGAMMA AND
C     CALCULATES THE STRESSES.

      REAL MU
      DIMENSICN FOFX(1)
      COMMCN
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     2/BLK14/YCR,IY
     7/BLK17/NEWR
     1/BLK20/IFIRST,IFAIL,NOCONV,NOS
      CALL EGAMMA(EX,EY,GAMXY,X,IY,ZU)
      SIGX = E/(1.-MU**2)*(EX+MU*EY)
      SIGY = E/(1.-MU**2)*(EY+MU*EX)
      TAUXY = E/(2.*(1.+MU))*GAMXY
      Y = SQRT(SIGX**2-SIGX*SIGY+SIGY**2+3.*TAUXY**2)
      IF(NEWR.EQ.1) PRINT 2, X,ZU,EX,EY,GAMXY,SIGX,SIGY,TAUXY,Y
    2 FORMAT(9E14.5)
      IF(Y.LE.YCR) GO TO 200
      IF(IFIRST.EQ.1) PRINT 10
   10 FORMAT(/* INITIAL DESIGN UNACCEPTABLE BECAUSE YIELD STRESS EXCEEDE
     1D*)
      IF(IFIRST.EQ.1) STOP
      IFAIL=1
      IF(IPRINT.NE.1) PRINT 11
   11 FORMAT(/* YIELD STRESS EXCEEDED - REDUCE S OR DC*)
      RETURN
  200 CONTINUE
      FOFX(1) = 1./(1.-Y/YCR)
      RETURN
      END
```

```
      SUBROUTINE EGAMMA(EX,EY,GAMXY,X,I,ZU)

C     CALCULATES STRAINS

      REAL MU
      DIMENSICN PHI(22)
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     3/BLK11/PHI
      CALL DEPTH(H,X)
      ZU = -H/2.
      I1 = I+NN1
      I2 = I+2*NN1
      EX = 2.*ZU*PHI(I2-1)
      IF(I.EC.1) EX = 0.
      IF(I-2)100,101,102
  100 EY = 2.*ZU/EPS**2*(PHI(I)+X*PHI(I1)+X**2*PHI(I2))
      GAMXY = 0.
      RETURN
  101 EY = ZU/EPS**2*(-2.*PHI(I-1)+PHI(I)+X*(-2.*PHI(I1-1)+PHI(I1))+
     1X**2*(-2.*PHI(I2-1)+PHI(I2)))
      GAMXY = ZU/EPS*(PHI(I1)+2.*X*PHI(I2))
      RETURN
  102 EY = ZU/EPS**2*(PHI(I-2)-2.*PHI(I-1)+PHI(I)+X*(PHI(I1-2)-2.*PHI(I1
     1-1)+PHI(I1))+X**2*(PHI(I2-2)-2.*PHI(I2-1)+PHI(I2)))
      GAMXY = ZU/EPS*(-PHI(I1-2)+PHI(I1)+2.*X*(-PHI(I2-2)+PHI(I2)))
      RETURN
      END
```

```
      SUBROUTINE FLUTTER
      REAL K,MU
      INTEGER ORD
      COMPLEX CDET,AAC
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     2/BLK2/CMEGA,V
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     4/BLK4/A,RHOA,IHP
     5/BLK5/CMEGA1,DELOM,OMFIN,V1,DELV,VFIN
     1/BLK6/IOPT,IPRINT,ISNDWCH
     8/BLK8/KDERIV
     7/BLK17/NEWR
     1/BLK20/IFIRST,IFAIL,NOCONV,NOS
     6/BLK26/OMSAVE,VSAVE
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLK31/DCR,OMPLUS,VPLUS
      KOUNT = 0
      NMAX=33
      OMEGA = OMSAVE = OMEGA1
      V = VSAVE = V1
      IF(NEWR.EQ.1) PRINT 61
   61 FORMAT(///10X*OMEGA*,17X*V*,24X*DETERMINANT*/
     17X*RADIANS/SEC.*,8X*INCHES/SEC.*,10X*REAL PART*,9X*IMAGINARY PART*
     2/)
 7702 CONTINUE

C     COMPLEX DETERMINANT

      CALL AAMAT
      CALL SMAT
      CALL PUTTOG
      ORD=NN33
      IF(IPRINT.EQ.0) RETURN
      DO 101 I=1,ORD
      DO 101 J=1,ORD
  101 AAC(I,J) = (1.E-15,0.)*AAC(I,J)
      CALL DEUPPS(AAC,ORD,CDET,NMAX)
      AO = REAL(CDET)
      BO = AIMAG(CDET)
      IF(NEWR.EQ.1) PRINT 6, CMEGA,V,CDET
    6 FORMAT(4E20.8)
```

```
      IF(IOPT.NE.1) GO TO 2000

C     ITERATE ON OMEGA AND V

      V = V+DELV
      IF(V.GT.VFIN) GO TO 7701
      GO TO 7702
 7701 OMEGA = OMEGA+DELOM
      IF(OMEGA.GT.OMFIN) STOP
      V = V1
      GO TO 7702
 2000 CONTINUE
      OMEGA = OMEGA1+DELOM
      CALL PUTTOG
      ORD=NN33
      DO 102 I=1,ORD
      DO 102 J=1,ORD
  102 AAC(I,J) = (1.E-15,0.)*AAC(I,J)
      CALL DEUPPS(AAC,ORD,CDET,NMAX)
      A1 = REAL(CDET)
      B1 = AIMAG(CDET)
      IF(NEWR.EQ.1) PRINT 6, OMEGA,V,CDET
      OMEGA = OMEGA1
      V = V1+DELV
      CALL PUTTOG
      ORD=NN33
      DO 103 I=1,ORD
      DO 103 J=1,ORD
  103 AAC(I,J) = (1.E-15,0.)*AAC(I,J)
      CALL DEUPPS(AAC,ORD,CDET,NMAX)
      A2 = REAL(CDET)
      B2 = AIMAG(CDET)
      IF(NEWR.EQ.1) PRINT 6, OMEGA,V,CDET
      H = (-A0*(B2-B0)/DELV+B0*(A2-A0)/DELV)/
     1((A1-A0)/DELOM*(B2-B0)/DELV-(B1-B0)/DELOM*(A2-A0)/DELV)
      K = (-B0*(A1-A0)/DELOM+A0*(B1-B0)/DELOM)/
     1((A1-A0)/DELOM*(B2-B0)/DELV-(B1-B0)/DELOM*(A2-A0)/DELV)
      OMEGA = OMEGA1+H
      V = V1+K
      CALL PUTTOG
      ORD=NN33
      DO 104 I=1,ORD
      DO 104 J=1,ORD
```

```
104 AAC(I,J) = (1.E-15,0.)*AAC(I,J)                                   17940000
    CALL DEUPPS(AAC,ORD,CDET,NMAX)                                    17950000
    AO = REAL(CDET)                                                   17960000
    BO = AIMAG(CDET)                                                  17970000
    IF(NEWR.EQ.1) PRINT 6, OMEGA,V,CDET                               17980000
    KOUNT = KOUNT+1                                                   17990000
    TEST = 1.E-5                                                      18000000
    IF(ABS((OMEGA1-OMEGA)/OMEGA).LT.TEST.AND.ABS((V1-V)/V).LT.TEST)   18010000
   1GO TO 300                                                         18020000
    IF(KOUNT.LT.7) GO TO 500                                          18030000
    GO TO(501,502),IFIRST                                             18040000
501 CONTINUE                                                          18050000
    PRINT 555                                                         18060000
555 FORMAT(/* 7 IMPROVEMENTS ON OMINIT AND VINIT WITHOUT CONVERGENCE*) 18070000
    STOP                                                              18080000
502 CONTINUE                                                          18090000
    IF(IPRINT.NE.1) PRINT 666                                         18100000
666 FORMAT(/* 7 IMPROVEMENTS, NO CONVERGENCE, REDUCE SINC*/)          18110000
    NOCONV = 1                                                        18120000
    OMEGA1 = OMSAVE                                                   18130000
    V1 = VSAVE                                                        18140000
    RETURN                                                            18150000
500 CONTINUE                                                          18160000
    OMEGA1 = OMEGA                                                    18170000
    V1 = V                                                            18180000
    GO TO 2000                                                        18190000
300 IF(NEWR.EQ.1) PRINT 301                                           18200000
301 FORMAT(/* COMPLEX DETERMINANT CONVERGED TO 0*)                    18210000
    Q = .5*RHOA*V**2*144.                                             18220000
    IF(IPRINT.EQ.3.OR.NEWR.EQ.1) PRINT 302, Q                         18230000
302 FORMAT(/* FLUTTER Q =*E16.8*   LB./SQ.FT.*)                       18240000
    OMPLUS = OMEGA                                                    18250000
    VPLUS = V                                                         18260000
    IF(KDERIV.EQ.0) GO TO 5000                                        18270000
    OMEGA1 = OMSAVE                                                   18280000
    V1 = VSAVE                                                        18290000
5000 RETURN                                                           18300000
    END                                                               18310000
```

```
      SUBROUTINE DEUPPS(A,N,DET,MAX)

C     EVALUATES COMPLEX DETERMINANT IN FLUTTER
C     DEUPPS = DETERMINANT EVALUATION USING PARTIAL PIVOTAL  STRATEGY

      COMPLEX A(MAX,N),SWAP,DET,C0,C1
      C0 = (0.,0.)
      C1 = (1.,0.)
      DET = C1
      NN = N-1

C     PIVOT SEARCH

      DO 560 I=1,NN
      CAVM = 0.
      II = I
      DO 105 J=II,N
      SWAP = A(J,II)
      AR = REAL(SWAP)
      AI = AIMAG(SWAP)
      CAVA = ABS(AR)+ABS(AI)
      IF(CAVM.GE.CAVA) GO TO 105
      IROW = J
      CAVM = CAVA
  105 CONTINUE
      IF(CAVM.EQ.0.) GO TO 720

C     ROW INTERCHANGE

      IF(IROW.EQ.II) GO TO 203
      DET = -DET
      DO 200 L=II,N
      SWAP = A(IROW,L)
      A(IROW,L) = A(II,L)
      A(II,L) = SWAP
  200 CONTINUE
  203 SWAP = A(II,II)
      DET = DET*SWAP

C     NORMALIZE PIVOT ROW

      K = II+1
```

```
      DO 350 L=K,N
  350 A(II,L) = A(II,L)/SWAP

C     ELIMINATION

      DO 550 L1=K,N
      SWAP = A(L1,II)
      DO 500 L=K,N
      A(L1,L) = A(L1,L)-A(II,L)*SWAP
  500 CONTINUE
  550 CONTINUE
  560 CONTINUE
      GO TO 730
  720 DET = C0
      GO TO 750
  730 DET = DET*A(N,N)
  750 RETURN
      END
```

```
      SUBROUTINE SMAT

C     GENERATES DIAGONAL ELEMENTS IN S MATRIX

      REAL MU
      COMPLEX AAC
      DIMENSION SUMS(5),FOFX(5)
      EXTERNAL FUNCS
      COMMON
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     3/BLK13/Y,C1,C2
      DO 100 I=1,NN
      Y = EPS*FLOAT(I)
      CALL CEE
      CALL MGAUSS(C1,C2,NCS,SUMS,FUNCS,FOFX,5)
      DO 50 K=1,5
      S(K,I)=SUMS(K)
      IF (I.EQ.NN) S(K,I)=.5*S(K,I)
   50 CONTINUE
  100 CONTINUE
      DO 200 K=1,5
  200 S(K,NN1)=0.
      IF(IPRINT.NE.0) RETURN
      PRINT 1, ((S(J,I),J=1,5),I=1,NN1)
    1 FORMAT(////* DIAGONAL ELEMENTS OF S MATRICES*/* PRINTED HERE AS CO
     1LUMNS SO-S4 (PROGRAM NOTATION IS S(1,I)-S(5,I)), ROWS ARE STATIONS
     1 Y=1,NN1*//(5E16.8))
      RETURN
      END
```

```
      SUBROUTINE FMAT
C    GENERATES DIAGONAL ELEMENTS IN F MATRIX

    1 FORMAT(1H1//* DIAGONAL ELEMENTS OF F MATRICES*/* PRINTED HERE AS C
     1OLUMNS F0-F4 (PROGRAM NOTATION IS F(1,I)-F(5,I))*/* ROWS ARE STATI
     1ONS Y=1,NN1*//(5E16.8))
      REAL MU
      COMPLEX AAC
      COMMON
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     1/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     1/BLK13/Y,C1,C2
      DO 100 I=1,NN
      Y=EPS*FLOAT(I)
      CALL CEE
      DO 50 K=1,5
      XK=K
      F(K,I)=1./XK*(C2**K-C1**K)
      IF(I.EQ.NN) F(K,I)=.5*F(K,I)
   50 CONTINUE
  100 CONTINUE
      DO 200 K=1,5
  200 F(K,NN1)=0.
      IF(IPRINT.NE.0) RETURN
      PRINT 1, ((F(J,I),J=1,5),I=1,NN1)
      RETURN
      END
```

```
      SUBROUTINE HMAT
C     GENERATES DIAGONAL ELEMENTS IN HP MATRIX

    1 FORMAT(/////* DIAGONAL ELEMENTS OF HP MATRICES*/* PRINTED HERE AS
     1COLUMNS HP1-HP4 (PROGRAM NOTATION IS HP(2,I)-HP(5,I)), ROWS ARE ST
     1ATIONS Y=1,NN1*/* HP0=0 AND IS NOT CALCULATED*//(4E16.8))
      REAL MU
      COMPLEX AAC
      COMMON
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     1/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK4/A,RHOA,IHP
     1/BLK6/IOPT,IPRINT,ISNDWCH
     1/BLK13/Y,C1,C2
     1/BLK18/DR
      DO 100 I=1,NN
      Y=EPS*FLOAT(I)
      CALL CEE
      IF(IHP.EQ.0) GO TO 150
      DO 120 N=2,5
      FN=FLOAT(N-1)
      NEW=N-1
      NEW2=NEW+2
      HP(N,I)=-8.*DR/(C2-C1)*(FN*(C2**NEW2-C1**NEW2)/(2.*(FN+2.)*(FN+1.)
     1)+C1*C2*(C1**NEW-C2**NEW)/(2.*(FN+1.)))
      IF(I.EQ.NN) HP(N,I)=.5*HP(N,I)
  120 CONTINUE
      GO TO 100
  150 CONTINUE
      HP(2,I)=HP(3,I)=HP(4,I)=HP(5,I)=0.
  100 CONTINUE
      DO 200 N=2,5
  200 HP(N,NN1)=0.
      IF(IPRINT.NE.0) RETURN
      PRINT 1, ((HP(J,I),J=2,5),I=1,NN1)
      RETURN
      END
```

```
      SUBROUTINE PUTTOG

    1 FORMAT(1H1//* AAC MATRIX*,5XI3,* X*I3,*  COMPLEX AA MATRIX AUGMENT
     1ED WITH INERTIA TERMS (S MATRICES) AND PISTON THEORY  (F AND HP AR
     2RAYS)*//)
    2 FORMAT(1X10E11.4)
    4 FORMAT(1H1/////)
   21 FORMAT(* PISTON THEORY USES OMEGA =*E16.8,*  AND V =*E16.8///)

      REAL MU
      COMPLEX AAC
      COMMON
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     2/BLK2/OMEGA,V
     1/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK4/A,RHOA,IHP
     1/BLK6/IOPT,IPRINT,ISNDWCH
      OM2=OMEGA**2
      EPS4=EPS**4
      GAM=1.4
      GAM14=(GAM+1.)/4.
      DO 100 I=1,NN33
      DO 100 J=1,NN33
  100 AAC(I,J)=AA(I,J)
      DO 200 I=1,NN1
      INN1=I+NN1
      INN22=I+NN22
      AREAL=AA(I,I)+EPS4*(-OM2*S(1,I))
      AMAG=EPS4*(2.*RHOA*A*OMEGA*F(1,I))
      AAC(I,I)=CMPLX(AREAL,AMAG)
      AREAL=AA(I,INN1)+EPS4*(-OM2*S(2,I)+2.*RHOA*A*V*F(1,I))
      AMAG=EPS4*(2.*RHOA*A*OMEGA*F(2,I)+2.*RHOA*V*GAM14*OMEGA*HP(2,I))
      AAC(I,INN1)=CMPLX(AREAL,AMAG)
      AREAL=AA(I,INN22)+EPS4*(-OM2*S(3,I)+4.*RHOA*A*V*F(2,I)+4.*RHOA*V
     1*V*GAM14*HP(2,I))
      AMAG=EPS4*(2.*RHOA*A*OMEGA*F(3,I)+2.*RHOA*V*GAM14*OMEGA*HP(3,I))
      AAC(I,INN22)=CMPLX(AREAL,AMAG)
      AREAL=AA(INN1,I)+EPS4*(-OM2*S(2,I))
      AMAG=EPS4*(2.*RHOA*A*OMEGA*(F(2,I)+V*GAM14/A*HP(2,I)))
      AAC(INN1,I)=CMPLX(AREAL,AMAG)
      AREAL=AA(INN1,INN1)+EPS4*(-OM2*S(3,I)+2.*RHOA*A*V*F(2,I)
```

```
      1+2.*RHOA*V*V*GAM14*HP(2,I))
       AMAG=EPS4*(2.*RHOA*A*OMEGA*F(3,I)+2.*RHOA*V*GAM14*OMEGA*HP(3,I))
       AAC(INN1,INN1)=CMPLX(AREAL,AMAG)
       AREAL=AA(INN1,INN22)+EPS4*(-OM2*S(4,I)+4.*RHOA*A*V*F(3,I)
      1+4.*RHCA*V*V*GAM14*HP(3,I))
       AMAG=EPS4*(2.*RHOA*A*OMEGA*F(4,I)+2.*RHOA*V*GAM14*OMEGA*HP(4,I))
       AAC(INN1,INN22)=CMPLX(AREAL,AMAG)
       AREAL=AA(INN22,I)+EPS4*(-CM2*S(3,I))
       AMAG=EPS4*(2.*RHOA*A*OMEGA*(F(3,I)+V*GAM14/A*HP(3,I)))
       AAC(INN22,I)=CMPLX(AREAL,AMAG)
       AREAL=AA(INN22,INN1)+EPS4*(-OM2*S(4,I)+2.*RHOA*A*V*F(3,I)
      1+2.*RHOA*V*V*GAM14*HP(3,I))
       AMAG=EPS4*(2.*RHOA*A*OMEGA*F(4,I)+2.*RHOA*V*GAM14*OMEGA*HP(4,I))
       AAC(INN22,INN1)=CMPLX(AREAL,AMAG)
       AREAL=AA(INN22,INN22)+EPS4*(-OM2*S(5,I)+4.*RHOA*A*V*F(4,I)
      1+4.*RHOA*V*V*GAM14*HP(4,I))
       AMAG=EPS4*(2.*RHOA*A*OMEGA*F(5,I)+2.*RHOA*V*GAM14*OMEGA*HP(5,I))
       AAC(INN22,INN22)=CMPLX(AREAL,AMAG)
   200 CONTINUE
       IF(IPRINT.NE.0) RETURN
       PRINT 1, NN33,NN33
       PRINT 21, OMEGA,V
       PRINT 2, ((AAC(I,J),J=1,NN1),I=1,NN33)
       PRINT 4
       PRINT 2, ((AAC(I,J),J=NN2,NN22),I=1,NN33)
       PRINT 4
       PRINT 2, ((AAC(I,J),J=NN23,NN33),I=1,NN33)
       RETURN
       END
```

```
      SUBROUTINE CEE                                                    20640C000
                                                                        206500000
C     DEFINES LEADING EDGE (C1) AND TRAILING EDGE (C2)                  2C660C000
                                                                        206700000
      COMMON                                                            206800000
     3/BLK36/C1A,C1B,C2A                                                206900000
     3/BLK13/Y,C1,C2                                                    207000000
      C1 = C1A+C1B*Y                                                    207100000
      C2 = C2A                                                          207200000
      RETURN                                                            207300000
      END                                                               207400000
```

```
      SUBROUTINE AAMAT

    1 FORMAT(1H1//* AA MATRIX*,5XI3,* X*I3,5X*    A MATRIX IN EQ. 29, PG.
     1 13 NACA TN-3640*//)
    2 FORMAT(5E16.8)
    5 FORMAT(1H1/////)
   10 FORMAT(1H1/* THE FOLLOWING MATRICES ARE PRINTED FROM SUBROUTINES C
     1ALMAT AND AMAT*////)

      DIMENSION A(11,11),AT(11,11)
      COMPLEX AAC
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     1/BLK6/IOPT,IPRINT,ISNOWCH
     1/NEWBLK/AAC(33,33),AA(33,33),S(5,11),F(5,11),HP(5,11)
      ITRANS = 0
      MOVE = 1

C     NA = 1      AA(1,1) TO AA(NN1,NN1) = A11
C     NA = 2      AA(1,NN2) TO AA(NN1,NN22) = A12
C                 AA(NN2,1) TO AA(NN22,NN1) = A21
C     NA = 3      AA(1,NN23) TO AA(NN1,NN33) = A13
C                 AA(NN23,1) TO AA(NN33,NN1) = A31
C     NA = 4      AA(NN2,NN2) TO AA(NN22,NN22) = A22
C     NA = 5      AA(NN2,NN23) TO AA(NN22,NN33) = A23
C                 AA(NN23,NN2) TO AA(NN34,NN23) = A32
C     NA = 6      AA(NN23,NN23) TO AA(NN33,NN33) = A33

      IF(IPRINT.EQ.0) PRINT 10
      DO 100 NA=1,6
      CALL AMAT(NA,A)
      IF(NA.EQ.4.OR.NA.EQ.5) GO TO 104
      IF(NA.EQ.6) GO TO 106
      DO 110 I=1,NN1
      DO 110 J=1,NN1
      JA = J
      IF(NA.EQ.2) JA = J+NN1
      IF(NA.EQ.3) JA = J+NN22
      AAC(I,JA) = AA(I,JA) = A(I,J)
  110 CONTINUE
      GO TO (100,200,200), NA
  104 DO 140 I=1,NN1
```

```
      DO 140 J=1,NN1
      JA = J
      IF(NA.EQ.4) JA=J+NN1
      IF(NA.EQ.5) JA=J+NN22
      IA = I+NN1
      AAC(IA,JA) = AA(IA,JA) = A(I,J)
  140 CONTINUE
      IF(NA.EQ.5) GO TO 200
      GO TO 100
  106 DO 160 I=1,NN1
      DO 160 J=1,NN1
      JA = J
      IF(NA.EQ.5) JA=J+NN1
      IF(NA.EQ.6) JA = J+NN22
      IA = I+NN22
      AAC(IA,JA) = AA(IA,JA) = A(I,J)
  160 CONTINUE
      GO TO 100
  200 CALL MATRIX(ITRANS,NN1,NN1,0,A,MCS,AT,MCS)
      CALL MATRIX(MOVE,NN1,NN1,0,AT,MCS,A,MCS)
      IF(NA.EQ.2) GO TO 104
      GO TO 106
  100 CONTINUE
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NN33,NN33
      PRINT 2, ((AA(I,J),J=1,NN1),I=1,NN33)
      PRINT 5
      PRINT 2, ((AA(I,J),J=NN2,NN22),I=1,NN33)
      PRINT 5
      PRINT 2, ((AA(I,J),J=NN23,NN33),I=1,NN33)
      RETURN
      END
```

```
      SUBROUTINE AMAT(NA,DUM1)

    1 FORMAT(//* AMAT     NA =*I3/)
    2 FORMAT(5E16.8)

      REAL MU
      DIMENSION DUM1(11,11),DUM2(11,11),DUM3(11,11)
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     2/BLK2/CMEGA,V
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
      MULT = 20
      ISCLR = 24
      ISUB = 22
      IADD = 21
      ITRANS = 0
      SCLR = 2.*MU*EPS**2
      SCLR2 = 2.*(1.-MU)*EPS**2
      SCLR4 = 4.*(1.-MU)*EPS**2
      SCLR8 = 8.*(1.-MU)*EPS**2
      SCLR4E = 4.*EPS**4
      GO TO (11,12,13,22,23,33),NA

C     A11 = D1*A1*D1T

   11 CALL CALMAT(1,1,DUM1,NN1,NN1)
      CALL CALMAT(3,1,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(2,1,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NA
      PRINT 2, ((DUM1(I,J),J=1,NN1),I=1,NN1)
      RETURN

C     A12 = D1*A2*D1T

   12 CALL CALMAT(1,1,DUM1,NN1,NN1)
      CALL CALMAT(3,2,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(2,1,DUM2,NN1,NN1)
```

```
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NA
      PRINT 2, ((DUM1(I,J),J=1,NN1),I=1,NN1)
      RETURN

C     A13 = D1*(A3*D1T+SCLR*A1BAR)

   13 CALL CALMAT(3,3,DUM1,NN1,NN1)
      CALL CALMAT(2,1,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(4,1,DUM2,NN1,NN1)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR,0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM2,MCS)
      CALL CALMAT(1,1,DUM1,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NA
      PRINT 2, ((DUM1(I,J),J=1,NN1),I=1,NN1)
      RETURN

C     A22 = D1*A3*D1T+SCLR*D4*A1STR*D4T

   22 CALL CALMAT(1,1,DUM1,NN1,NN1)
      CALL CALMAT(3,3,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(2,1,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(1,4,DUM2,NN1,NN)
      CALL CALMAT(5,1,DUM3,NN,NN)
      CALL MATRIX(MULT,NN1,NN,NN,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL CALMAT(2,4,DUM3,NN1,NN)
      CALL MATRIX(MULT,NN1,NN,NN1,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR2,0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NA
      PRINT 2, ((DUM1(I,J),J=1,NN1),I=1,NN1)
      RETURN

C     A23 = D1*A4*D1T+SCLR4*D4*A2STR*D4T+SCLR*D1*A2BAR

   23 CALL CALMAT(1,1,DUM1,NN1,NN1)
```

```
      CALL CALMAT(3,4,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(2,1,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(1,4,DUM2,NN1,NN)
      CALL CALMAT(5,2,DUM3,NN,NN)
      CALL MATRIX(MULT,NN1,NN,NN,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL CALMAT(2,4,DUM3,NN1,NN)
      CALL MATRIX(MULT,NN1,NN,NN1,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR4,0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(1,1,DUM2,NN1,NN1)
      CALL CALMAT(4,2,DUM3,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR,0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NA
      PRINT 2, ((DUM1(I,J),J=1,NN1),I=1,NN1)
      RETURN

C     A33 = D1*A5*D1T+SCLR8*D4*A3STR*D4T+SCLR*(D1*A3BAR+(D1*A3BAR)T)
C            +SCLR4E*A1HAT

   33 CALL CALMAT(1,1,DUM1,NN1,NN1)
      CALL CALMAT(3,5,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(2,1,DUM2,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(1,4,DUM2,NN1,NN)
      CALL CALMAT(5,3,DUM3,NN,NN)
      CALL MATRIX(MULT,NN1,NN,NN,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL CALMAT(2,4,DUM3,NN1,NN)
      CALL MATRIX(MULT,NN1,NN,NN1,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR8,0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      CALL CALMAT(1,1,DUM2,NN1,NN1)
      CALL CALMAT(4,3,DUM3,NN1,NN1)
      CALL MATRIX(MULT,NN1,NN1,NN1,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL MATRIX(ITRANS,NN1,NN1,0,DUM2,MCS,DUM3,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM2,MCS,DUM3,MCS,DUM2,MCS)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR,0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM1,MCS)
```

```
      CALL CALMAT(6,1,DUM2,NN1,NN1)
      CALL MATRIX(ISCLR,NN1,NN1,0,SCLR4E.0,DUM2,MCS,DUM2,MCS)
      CALL MATRIX(IADD,NN1,NN1,0,DUM1,MCS,DUM2,MCS,DUM1,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NA
      PRINT 2, ((DUM1(I,J),J=1,NN1),I=1,NN1)
      RETURN
      END
```

```
      SUBROUTINE MINGAGE(G3)
C     EVALUATES MINIMUM GAGE CONSTRAINT
      EXTERNAL FTHICK
      REAL MU
      DIMENSION TX(11),FOFX(1)
      DIMENSION ANS(1)
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK13/Y,C1,C2
     1/BLK20/IFIRST,IFAIL,NOCONV,NCS
     1/BLKNPS/AREA
     1/NPSBLK/ITHICK
      IF(ITHICK.EQ.1) INDEX=NN1
      IF(ITHICK.EQ.2) INDEX=NN
      DO 500 I=1,INDEX
      Y = EPS*FLOAT(I-1)
      CALL CEE
      CALL MGAUSS(C1,C2,NCS,ANS,FTHICK, FOFX,1)
      IF(IFAIL.EQ.1) RETURN
  500 TX(I) = ANS(1)
      IF(ITHICK.EQ.1) SUM=1./2.*TX(1)+1./2.*TX(NN1)
      IF(ITHICK.EQ.2) SUM=1./2.*TX(1)
      DO 200 I=2,NN
  200 SUM = SUM+TX(I)
      SUM = SUM*EPS/AREA
      G3 = 1.-1./SUM
      RETURN
      END
```

```
      SUBROUTINE FTHICK(X,FOFX)
      DIMENSION FOFX(1)
      COMMCN
     1/BLK6/IOPT,IPRINT,ISNCWCH
     7/BLK7/TBAR
     7/BLK17/NEWR
     1/BLK20/IFIRST,IFAIL,NOCONV,NOS
     3/BLK25/TBARMIN
      CALL THKNESS(X)
      IF(TBAR.GE.TBARMIN) GO TO 200
      IF(IFIRST.EQ.1) PRINT 10
   10 FORMAT(/* INITIAL DESIGN UNACCEPTABLE BECAUSE THICKNESS IS LESS TH
     1AN MINIMUM GAGE*)
      IF(IFIRST.EQ.1) STOP
      IFAIL=1
      IF(IPRINT.EQ.3.OR.NEWR.EQ.1) PRINT 11
   11 FORMAT(/* THICKNESS LESS THAN MINIMUM GAGE - REDUCE S OR DC*/)
      RETURN
  200 CONTINUE
      FOFX(1) = 1./(1.-TBARMIN/TBAR)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE THKNESS(X)
      REAL MU
      COMMON
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK13/Y,C1,C2
     7/BLK7/TBAR
     1/NPSBLK/ITHICK
      A = 1.-Y/SS
      B = 1.-2.*X/(C1+C2)
      IF(ITHICK.EQ.2) GO TO 100
      TBAR =    C( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3
     1        +(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B
     1        +(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*B**2
     1        +(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*B**3
      RETURN
  100 CONTINUE
      TBAR =    C( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3
     1        +(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B
     1        +(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*(1-B*B)
     1        +(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*(1-B*B)*B
      RETURN
      END
```

```
      SUBROUTINE CALMAT(NAME,NSUB,DUM,IRO ,ICO )                         236100000
                                                                         236200000
    1 FORMAT(//* NAME =*I3,*   NSUB =*I3/)                               236300000
    2 FORMAT(5E16.8)                                                     236400000
    3 FORMAT(4E16.8)                                                     236500000
    4 FORMAT(6E16.8)                                                     236600000
                                                                         236700000
      DIMENSICN FOFX(1)                                                  236800000
      DIMENSICN SUMA(1),SUMS(1)                                          236900000
      DIMENSICN DUM(11,11),D(11,11),DT(11,11),A(11,11)                   237000000
      REAL MU                                                            237100000
      EQUIVALENCE (D,DT,A)                                               237200000
      EXTERNAL FUNCA                                                     237300000
      COMMON                                                             237400000
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX                       237500000
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS                                   237600000
     1/BLK6/IOPT,IPRINT,ISNDWCH                                          237700000
     9/BLK9/NSUBX                                                        237800000
     1/BLK13/Y,C1,C2                                                     237900000
                                                                         238000000
C     NAME = 1   D                                                       238100000
C     NAME = 2   DT                                                      238200000
C     NAME = 3   A                                                       238300000
C     NAME = 4   ABAR                                                    238400000
C     NAME = 5   ASTAR                                                   238500000
C     NAME = 6   AHAT                                                    238600000
                                                                         238700000
      NSUBX=NSUB                                                         238800000
      IF(NAME.GT.2) GO TO 300                                            238900000
      IF(NSUB.EQ.4) GO TO 140                                            239000000
                                                                         239100000
C     CALCULATE D1 MATRIX                                                239200000
                                                                         239300000
  110 DO 111 I=1,IRO                                                     239400000
      DO 111 J=1,ICO                                                     239500000
      D(I,J) = 0.                                                        239600000
      IF(J.EQ.I) D(I,J) = 1.                                             239700000
      IF(J.EQ.I+1) D(I,J) = -2.                                          239800000
      IF(J.EQ.I+2) D(I,J) = 1.                                           239900000
  111 CONTINUE                                                           240000000
      D(1,1) = 2.                                                        240100000
      IF(NAME.EQ.2) GO TO 200                                            240200000
```

```
      IF(IPRINT.NE.0) GO TO 500
      PRINT 1, NAME,NSUB
      PRINT 2, ((D(I,J),J=1,ICO ),I=1,IRO )
      GO TO 500

C     CALCULATE D4 MATRIX

  140 DO 141 I=1,IRO
      DO 141 J=1,ICO
      D(I,J) = 0.
      IF(J.EQ.I) D(I,J) = 1.
      IF(J.EQ.I+1) D(I,J) = -1.
  141 CONTINUE
      IF(NAME.EQ.2) GO TO 200
      IF(IPRINT.NE.0) GO TO 500
      PRINT 1, NAME,NSUB
      PRINT 3, ((D(I,J),J=1,ICO ),I=1,IRO )
      GO TO 500
  200 CONTINUE

C     CALCULATE DT MATRIX

      ITRANS = 0
      CALL MATRIX(ITRANS,IRO ,ICO ,0,D,MCS,DUM,MCS)
      IF(IPRINT.NE.0) RETURN
      PRINT 1, NAME,NSUB
      DO 31 I=1,ICO
   31 PRINT 4, (DUM(I,J),J=1,IRO)
      RETURN
  300 CONTINUE

C     CALCULATE A MATRIX

      DO 301 I=1,IRO
      DO 301 J=1,ICO
      A(I,J) = 0.
      IF(J.NE.I) GO TO 301
      N = J-1
      Y = EPS*FLOAT(N)
      IF(NAME.EQ.5) Y = EPS*(FLOAT(N)+1./2.)
      CALL CEE
      CALL MGAUSS(C1,C2,NCS,SUMA,FUNCA,FOFX,1)
      A(I,J) = SUMA(1)
```

```
      IF(NAME.EQ.5) GO TO 301
      IF(I.EQ.1.OR.I.EQ.NN1) A(I,J) = 1./2.*A(I,J)
  301 CONTINUE
      IF(NAME.EQ.4) GO TO 310
      IF(NAME.EQ.6) GO TO 320
      GO TO 400

C     SHIFT TO FORM ABAR MATRIX

  310 DO 312 I=2,IRO
      J=I-1
      A(I,J)=A(I,I)
  312 A(I,I)=0.
      A(1,1)=0.
      GO TO 400

C     SHIFT TO FORM AHAT MATRIX

  320 IRO1=IRO-1
      DO 322 I=1,IRO1
  322 A(I,I)=A(I+1,I+1)
      A(IRO,IRO)=0.
  400 CONTINUE
      IF(IPRINT.NE.0) GO TO 500
      PRINT 1, NAME,NSUB
      DO 41 I=1,IRO
   41 PRINT 4, (A(I,J),J=1,ICO )

  500 MOVE = 1
      CALL MATRIX(MOVE,IRO ,ICO ,0,D,MCS,DUM,MCS)
      RETURN
      END
```

```
      SUBROUTINE FUNCA(X,FOFX)

C     EVALUATES BENDING STIFFNESS D

      DIMENSION FOFX(1)
      REAL MU
      COMMCN
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNOWCH
     7/BLK7/TBAR
     9/BLK9/NSUB
      CALL THKNESS(X)
      IF(ISNOWCH.EQ.1) GO TO 100
      D = E*TBAR**3/(12.*(1.-MU**2))
      GO TO 200
  100 CALL DEPTH(H,X)
      D = E*H**2/2.*TBAR/(1.-MU**2)+2.*(E*TBAR**3/(12.*(1.-MU**2)))
  200 CONTINUE
      FOFX(1) = D*X**(NSUB-1)
      RETURN
      END
```

```
      SUBROUTINE FUNCS(X,FOFX)

C   EVALUATES MASS PER UNIT AREA M

      REAL M
      REAL MU
      DIMENSION FOFX(1)
      COMMCN
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     1/BLK6/IOPT,IPRINT,ISNDWCH
     7/BLK7/TBAR
     1/BLK50/RHOF
      CALL THKNESS(X)
      IF(ISNDWCH.EQ.1) GO TO 100
      M = RHCW*TBAR
      GO TC 200
  100 CALL DEPTH(H,X)
      M = 2.*RHOW*TBAR+RHOF*H
  200 CONTINUE
      DO 300 I=1,5
  300 FOFX(I)=M*X**(I-1)
      RETURN
      END
```

```
      SUBROUTINE DEPTH(H,X)

C   AIRFOIL MUST BE SYMMETRIC - THAT IS, NO CAMBER.

      COMMON
     1/BLK13/Y,C1,C2
     1/BLK6/IOPT,IPRINT,ISNDWCH
     7/BLK7/TBAR
     8/BLK18/D
      IF(ISNDWCH.EQ.0) GO TO 100
      H=D*(C2-C1)-4.*D/(C2-C1)*(X-(C2+C1)/2.)**2
      RETURN
  100 CALL THKNESS(X)
      H = TBAR
      RETURN
      END
```

```
      SUBROUTINE TABLE

    1 FORMAT(1H1///* THICKNESS TABLE*//
     1*    Y             X                         TBAR*//)
    2 FORMAT(F6.0,2E20.8)

      REAL MU
      COMMON
     1/BLK1/NN,NN1,NN2,NN22,NN23,NN33,MCS,NCS,NMAX
     3/BLK3/ANG,RHOW,E,MU,C(16),SS,EPS
     7/BLK7/TBAR
     3/BLK13/Y,C1,C2
      PRINT 1
      YINC = SS/(2.*FLOAT(NN))
      Y = 0.
  200 CONTINUE
      CALL CEE
      XINC = (C2-C1)/(20.*FLOAT(NCS))
      X = C1
  100 CONTINUE
      CALL THKNESS(X)
      PRINT 2, Y,X,TBAR
      X = X+XINC
      IF(X.GT.(C2+.0001)) GO TO 101
      GO TO 100
  101 CONTINUE
      Y = Y+YINC
      IF(Y.GT.SS) RETURN
      GO TO 200
      END
```

```
      SUBROUTINE ITR2 (X,A,B,DELTX,FOFX,E1,E2,MAXI,ICODE)               ITR2  256800000
********* DOCUMENT DATE 08-01-68   SUBROUTINE REVISED 08-01-68 ********* ITR2  256900000
      X=A                                                               ITR2  257000000
      KX=0                                                              ITR2  257100000
      LX=0                                                              ITR2  257200000
      IF (DELTX)111,111,112                                             ITR2  257300000
  112 IF (B- A ) 113,113,114                                            ITR2  257400000
  114 I =0                                                              ITR2  257500000
      IF (FOFX(A))1,2,3                                                 ITR2  257600000
    1 XB1=X                                                             ITR2  257700000
      IF(LX.NE.0)GO TO 1001                                             ITR2  257800000
      X =X+DELTX                                                        ITR2  257900000
      IF(X-B)1000,1000,1004                                             ITR2  258000000
 1004 X=B                                                               ITR2  258100000
      LX=1                                                              ITR2  258200000
 1000 IF (FOFX(X))1,2,4                                                 ITR2  258300000
    4 XB=X                                                              ITR2  258400000
      X=X-DELTX/(2.**(I+1))                                             ITR2  258500000
  999 I=I+1                                                             ITR2  258600000
      IF(MAXI.LT.I)GO TO 444                                            ITR2  258700000
      IF (FOFX(X))6,2,7                                                 ITR2  258800000
    6 L=1                                                               ITR2  258900000
      XX=XB                                                             ITR2  259000000
      GO TO 18                                                          ITR2  259100000
    7 L=2                                                               ITR2  259200000
      XX=XB1                                                            ITR2  259300000
      GO TO 18                                                          ITR2  259400000
    3 XB1=X                                                             ITR2  259500000
      IF(KX.NE.0)GO TO 1001                                             ITR2  259600000
      X= X+DELTX                                                        ITR2  259700000
      IF(X-B)1002,1002,1003                                             ITR2  259800000
 1003 X=B                                                               ITR2  259900000
      KX=1                                                              ITR2  260000000
 1002 IF(FOFX(X))5,2,3                                                  ITR2  260100000
    5 XB=X                                                              ITR2  260200000
      X=X-DELTX/(2.**(I+1))                                             ITR2  260300000
  998  I=I+1                                                            ITR2  260400000
      IF(MAXI.LT.I)GO TO 444                                            ITR2  260500000
      IF(FOFX(X))8,2,9                                                  ITR2  260600000
    9 L=3                                                               ITR2  260700000
      XX=XB                                                             ITR2  260800000
      GO TO 18                                                          ITR2  260900000
```

```
    8 L =4                                                  ITR2    261000000
      XX=XB1                                                ITR2    261100000
   18 IF (ABS(X)-E1)36,36,37                                ITR2    261200000
   37 IF (ABS((XX-X)/X)-E1)2,2,17                           ITR2    261300000
   36 IF (ABS(XX-X)-E2)2,2,17                               ITR2    261400000
   17 GO TO (81,4,81,5),L                                   ITR2    261500000
   81 XB1 =X                                                ITR2    261600000
      X=X+DELTX/(2.**(I+1))                                 ITR2    261700000
      GO TO (999,4,998,5),L                                 ITR2    261800000
  111 ICODE =2                                              ITR2    261900000
      GO TO 79                                              ITR2    262000000
  113 ICODE =4                                              ITR2    262100000
      GO TO 79                                              ITR2    262200000
 1001 ICODE =3                                              ITR2    262300000
      GO TO 79                                              ITR2    262400000
  444 ICODE= 1                                              ITR2    262500000
      GO TO 79                                              ITR2    262600000
    2 ICODE =0                                              ITR2    262700000
   79 CONTINUE                                              ITR2    262800000
      RETURN                                                ITR2    262900000
      END                                                   ITR2    263000000
```

```
      SUBROUTINE SIMEQ(A,N,B,M,DETERM,IPIVOT,NMAX,ISCALE)               SIMEQ   263100000
C     SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS                         SIMEQ   263200000
********* DOCUMENT DATE 08-01-68   SUBROUTINE REVISED 08-01-68 ********* SIMEQ   263300000
C                                                                       SIMEQ   263400000
      DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M)                           SIMEQ   263500000
      EQUIVALENCE (IROW,JROW),(ICOLUM,JCOLUM),(AMAX,T,SWAP)             SIMEQ   263600000
C                                                                       SIMEQ   263700000
C     INITIALIZATION                                                    SIMEQ   263800000
C                                                                       SIMEQ   263900000
    5 ISCALE=0                                                          SIMEQ   264000000
    6 R1=10.0**100                                                      SIMEQ   264100000
    7 R2=1.0/R1                                                         SIMEQ   264200000
   10 DETERM=1.0                                                        SIMEQ   264300000
   15 DO 20 J=1,N                                                       SIMEQ   264400000
   20 IPIVOT(J)=0                                                       SIMEQ   264500000
   30 DO 550 I=1,N                                                      SIMEQ   264600000
C                                                                       SIMEQ   264700000
C     SEARCH FOR PIVOT ELEMENT                                          SIMEQ   264800000
C                                                                       SIMEQ   264900000
   40 AMAX=0.0                                                          SIMEQ   265000000
   45 DO 105 J=1,N                                                      SIMEQ   265100000
   50 IF (IPIVOT(J)-1) 60,105,60                                        SIMEQ   265200000
   60 DO 100 K=1,N                                                      SIMEQ   265300000
   70 IF (IPIVOT(K)-1) 80,100,740                                       SIMEQ   265400000
   80 IF (ABS(AMAX)-ABS(A(J,K))) 85,100,100                             SIMEQ   265500000
   85 IROW=J                                                            SIMEQ   265600000
   90 ICOLUM=K                                                          SIMEQ   265700000
   95 AMAX=A(J,K)                                                       SIMEQ   265800000
  100 CONTINUE                                                          SIMEQ   265900000
  105 CONTINUE                                                          SIMEQ   266000000
      IF (AMAX) 110,106,110                                             SIMEQ   266100000
  106 DETERM=0.0                                                        SIMEQ   266200000
      ISCALE=0                                                          SIMEQ   266300000
      GO TO 740                                                         SIMEQ   266400000
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1                                   SIMEQ   266500000
C                                                                       SIMEQ   266600000
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                 SIMEQ   266700000
C                                                                       SIMEQ   266800000
  130 IF (IROW-ICOLUM) 140,260,140                                      SIMEQ   266900000
  140 DETERM=-DETERM                                                    SIMEQ   267000000
  150 DO 200 L=1,N                                                      SIMEQ   267100000
  160 SWAP=A(IROW,L)                                                    SIMEQ   267200000
```

```
  170 A(IROW,L)=A(ICOLUM,L)                                     SIMEQ   267300000
  200 A(ICOLUM,L)=SWAP                                          SIMEQ   267400000
  205 IF (M) 260,260,210                                        SIMEQ   267500000
  210 DO 250 L=1,M                                              SIMEQ   267600000
  220 SWAP=B(IROW,L)                                            SIMEQ   267700000
  230 B(IROW,L)=B(ICOLUM,L)                                     SIMEQ   267800000
  250 B(ICOLUM,L)=SWAP                                          SIMEQ   267900000
  260 PIVOT=A(ICOLUM,ICOLUM)                                    SIMEQ   268000000
      IF (PIVOT) 1000,106,1000                                  SIMEQ   268100000
C                                                               SIMEQ   268200000
C     SCALE THE DETERMINANT                                     SIMEQ   268300000
C                                                               SIMEQ   268400000
 1000 PIVOTI=PIVOT                                              SIMEQ   268500000
 1005 IF(ABS(DETERM)-R1)1030,1010,1010                          SIMEQ   268600000
 1010 DETERM=DETERM/R1                                          SIMEQ   268700000
      ISCALE=ISCALE+1                                           SIMEQ   268800000
      IF(ABS(DETERM)-R1)1060,1020,1020                          SIMEQ   268900000
 1020 DETERM=DETERM/R1                                          SIMEQ   269000000
      ISCALE=ISCALE+1                                           SIMEQ   269100000
      GO TO 1060                                                SIMEQ   269200000
 1030 IF(ABS(DETERM)-R2)1040,1040,1060                          SIMEQ   269300000
 1040 DETERM=DETERM*R1                                          SIMEQ   269400000
      ISCALE=ISCALE-1                                           SIMEQ   269500000
      IF(ABS(DETERM)-R2)1050,1050,1060                          SIMEQ   269600000
 1050 DETERM=DETERM*R1                                          SIMEQ   269700000
      ISCALE=ISCALE-1                                           SIMEQ   269800000
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070                          SIMEQ   269900000
 1070 PIVOTI=PIVOTI/R1                                          SIMEQ   270000000
      ISCALE=ISCALE+1                                           SIMEQ   270100000
      IF(ABS(PIVOTI)-R1)320,1080,1080                           SIMEQ   270200000
 1080 PIVOTI=PIVOTI/R1                                          SIMEQ   270300000
      ISCALE=ISCALE+1                                           SIMEQ   270400000
      GO TO 320                                                 SIMEQ   270500000
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320                           SIMEQ   270600000
 2000 PIVOTI=PIVOTI*R1                                          SIMEQ   270700000
      ISCALE=ISCALE-1                                           SIMEQ   270800000
      IF(ABS(PIVOTI)-R2)2010,2010,320                           SIMEQ   270900000
 2010 PIVOTI=PIVOTI*R1                                          SIMEQ   271000000
      ISCALE=ISCALE-1                                           SIMEQ   271100000
  320 DETERM=DETERM*PIVOTI                                      SIMEQ   271200000
C                                                               SIMEQ   271300000
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                         SIMEQ   271400000
C                                                               SIMEQ   271500000
```

```
  340 DO 351 L=1,N                                                    SIMEQ   271600000
  341 IF (IPIVOT(L)-1) 350,351,740                                    SIMEQ   271700000
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT                                   SIMEQ   271800000
  351 CONTINUE                                                        SIMEQ   271900000
  355 IF (M) 380,380,360                                              SIMEQ   272000000
  360 DO 370 L=1,M                                                    SIMEQ   272100000
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT                                   SIMEQ   272200000
C                                                                     SIMEQ   272300000
C     REDUCE NON-PIVOT ROWS                                           SIMEQ   272400000
C                                                                     SIMEQ   272500000
  380 DO 550 L1=1,N                                                   SIMEQ   272600000
  390 IF (L1-ICOLUM) 400,550,400                                      SIMEQ   272700000
  400 T=A(L1,ICOLUM)                                                  SIMEQ   272800000
  430 DO 451 L=1,N                                                    SIMEQ   272900000
  431 IF (IPIVOT(L)-1) 450,451,740                                    SIMEQ   273000000
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                   SIMEQ   273100000
  451 CONTINUE                                                        SIMEQ   273200000
  455 IF (M) 550,550,460                                              SIMEQ   273300000
  460 DO 500 L=1,M                                                    SIMEQ   273400000
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T                                   SIMEQ   273500000
  550 CONTINUE                                                        SIMEQ   273600000
  740 RETURN                                                          SIMEQ   273700000
      END                                                             SIMEQ   273800000
```

The following is a listing of input data and a printout from a computer run used to determine the flutter-solution neighborhood:

```
 TITANIUM
$OPTION          IOPT=1, IPRINT=2,       ISNDWCH=1, ITHICK=2$
$WING ANG=50.5,RHOW=4.14E-4, E=1.64E7, MU=.3,
        AREA=5.65808E5, DR=3.E-2,
AR=2.5538,
         WTF = 150000.,
DC(1) = 9*0.,  3*1.E-6, 1*0., 3*1.E-6,      NC=16,
 C(1)=.02, 8*0., .23, 6*0.$
$AIR A=1.2204E4, RHOA=5.14E-8, CORALT=2.5E4, IHP=1$
$CASE NN=6,  NCS=1$
$FLUTER
OMINIT=0., DELOM=4., OMFIN=40.,    VINIT=0., DELV=1.E4, VFIN=1.E5$
$OPTIMUM R=300.,RDC=15.,VCR=2.5E4$
$STRSS  PLOAD=1., YCR=1.25E5$
$GAGE TBARMIN=.02 $
```

```
DETERMINATION OF THICKNESS DISTRIBUTION FOR MINIMUM WEIGHT WING
SATISFYING PRESCRIBED FLUTTER AND STRENGTH CRITERIA.
REFERENCE - NACA TN-3640,   STEIN-SANDERS,   JUNE,1956.
DEXTER-STROUD     RDK288     MAY,1969.


DATE 04/27/71


WING DESCRIPTION

    ASPECT RATIO............AR =  2.55380000E+00
    SURFACE AREA..........AREA =  5.65808000E+05 IN.**2
    L.E. SWEEP ANGLE.......ANG =  5.05000000E+01 DEGREES
    SEMISPAN................SS =  8.49988374E+02 IN.
    ROOT CHORD..............RC =  1.18122475E+03 IN.
    DEPTH RATIO.............DR =  3.00000000E-02
    WING MATERIAL.............   TITANIUM
    MASS DENSITY..........RHOW =  4.14000000E-04 (LB.*SEC.**2)/(IN.**4)
    YOUNGS MODULUS...........E =  1.64000000E+07 PSI
    POISSONS RATIO..........MU =  3.00000000E-01


AIR PROPERTIES

    SPEED OF SOUND...........A =  1.22040000E+04 IN./SEC. =  1.01700000E+03 FT./SEC.
    MASS DENSITY..........RHOA =  5.14000000E-08 (LB.*SEC.**2)/(IN.**4)
    CORRESPONDING ALTITUDE.... =  2.50000000E+04 FT.


FUEL PROPERTIES

    WEIGHT OF FUEL..........WTF =  1.50000000E+05 LB.
    VOLUME OF WING..........VCL =  9.03896471E+06 IN.**3
    DENSITY OF FUEL........RHOF =  4.29917638E-05 (LB.*SEC.**2)/(IN.**4)


EQUATION FOR THICKNESS - TBAR    (INCHES)

TBAR =    C( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3
        +(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B
        +(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*(1-B*B)
        +(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*(1-B*B)*B

           WHERE  A = 1-Y/SS   AND  B = 1-2*X/(LE+TE)

                       (ITHICK = 2)
```

```
                  XXX
                  XXX
                  XXX          FIXED BOUNDARY
                 .XXX          CONDITION
                . XXX
               .  XXX
              .   XXX
             .    XXX
            .     XXX
           .      XXX
          .       XXX  ROOT CHORD =1181.2 IN.
         .        XXX
        .         XXX
       .          XXX
      .           XXX
 . 50 DEG.        XXX
 . . . . . . . . .XXX
                  XXX
SEMISPAN =        XXX LEADING EDGE = A+B*Y
850.0 IN.         XXX         A = -1.18122475E+03
                              B =  1.21309700E+00
```

```
DESIGN VARIABLES                 FINITE DIFFERENCE INCREMENTS    ANALYTICAL DERIVATIVES        * INDICATES ACTIVE
                                                                                                 DESIGN VARIABLES
   C( 1) =  2.00000000E-02       DC( 1) =  0.                    AD( 1) =  1.80836763E+05
   C( 2) =  0.                   DC( 2) =  0.                    AD( 2) =  1.13761448E+05
   C( 3) =  0.                   DC( 3) =  0.                    AD( 3) =  8.36219870E+04
   C( 4) =  0.                   DC( 4) =  0.                    AD( 4) =  6.62179501E+04
   C( 5) =  0.                   DC( 5) =  0.                    AD( 5) =  0.
   C( 6) =  0.                   DC( 6) =  0.                    AD( 6) =  0.
   C( 7) =  0.                   DC( 7) =  0.                    AD( 7) =  0.
   C( 8) =  0.                   DC( 8) =  0.                    AD( 8) =  0.
   C( 9) =  0.                   DC( 9) =  0.                    AD( 9) =  6.02789211E+04
 * C(10) =  2.30000000E-01       DC(10) =  1.00000000E-06        AD(10) =  7.58409650E+04
 * C(11) =  0.                   DC(11) =  1.00000000E-06        AD(11) =  5.57479913E+04
 * C(12) =  0.                   DC(12) =  1.00000000E-06        AD(12) =  4.41453001E+04
   C(13) =  0.                   DC(13) =  0.                    AD(13) =  0.
 * C(14) =  0.                   DC(14) =  1.00000000E-06        AD(14) =  0.
 * C(15) =  0.                   DC(15) =  1.00000000E-06        AD(15) =  0.
 * C(16) =  0.                   DC(16) =  1.00000000E-06        AD(16) =  0.

STATIONS

   NUMBER OF SEMISPAN STATIONS..................NN =  6
   NUMBER OF CHORDWISE STATIONS = (10 X NCS)...NCS =  1

INPUT OPTIONS

     IOPT = 1     1 - ITERATE ON OMEGA AND V, 2 - COMPLEX SEARCH
   IPRINT = 2     0 - MATRIX CONSTRUCTION, 1 - SHORT PRINTOUT, 2 - INTERMEDIATE PRINTOUT, 3 - EXTENDED PRINTOUT
  ISNDWCH = 1     0 - SOLID WING, 1 - SANDWICH CONSTRUCTION
      IHP = 1     0 - THICKNESS VARIATION EFFECTS NOT ACCOUNTED FOR IN PISTON THEORY, 1 - THICKNESS EFFECTS ARE USED


INITIAL WEIGHT =  2.10601572E+04   POUNDS


RANGE AND INCREMENT OF ITERATION

   OMEGA INITIAL.............OMINIT =  0.              RADIANS/SEC.
   OMEGA INCREMENT............DELOM =  4.00000000E+00 RADIANS/SEC.
   OMEGA FINAL................OMFIN =  4.00000000E+01 RADIANS/SEC.
```

```
V INITIAL..................VINIT =  0.              IN./SEC.
V INCREMENT................DELV  =  1.00000000E+04  IN./SEC.
V FINAL....................VFIN  =  1.00000000E+05  IN./SEC.
```

OMEGA RADIANS/SEC.	V INCHES/SEC.	DETERMINANT REAL PART	DETERMINANT IMAGINARY PART
0.	0.	1.35385820E+03	0.
0.	1.00000000E+04	6.23527272E+03	0.
0.	2.00000000E+04	1.75389121E+04	0.
0.	3.00000000E+04	3.89624065E+04	0.
0.	4.00000000E+04	7.45373284E+04	0.
0.	5.00000000E+04	1.28062608E+05	0.
0.	6.00000000E+04	2.02432381E+05	0.
0.	7.00000000E+04	2.98951660E+05	0.
0.	8.00000000E+04	4.16746162E+05	0.
0.	9.00000000E+04	5.52370354E+05	0.
0.	1.00000000E+05	6.99700098E+05	0.
4.00000000E+00	0.	8.70789548E+02	4.04729382E+02
4.00000000E+00	1.00000000E+04	4.86932933E+03	1.06019211E+03
4.00000000E+00	2.00000000E+04	1.46321133E+04	2.20877611E+03
4.00000000E+00	3.00000000E+04	3.37252092E+04	3.99144196E+03
4.00000000E+00	4.00000000E+04	6.61215526E+04	6.51589754E+03
4.00000000E+00	5.00000000E+04	1.15649561E+05	9.83249378E+03
4.00000000E+00	6.00000000E+04	1.85321974E+05	1.39143444E+04
4.00000000E+00	7.00000000E+04	2.76639463E+05	1.86451088E+04
4.00000000E+00	8.00000000E+04	3.88978936E+05	2.38170344E+04
4.00000000E+00	9.00000000E+04	5.19175443E+05	2.91406108E+04
4.00000000E+00	1.00000000E+05	6.61388855E+05	3.42657030E+04
8.00000000E+00	0.	-1.19012943E+02	4.13508408E+02
8.00000000E+00	1.00000000E+04	1.79992767E+03	1.28275018E+03
8.00000000E+00	2.00000000E+04	7.79912174E+03	2.93147930E+03
8.00000000E+00	3.00000000E+04	2.09783986E+04	5.63686043E+03
8.00000000E+00	4.00000000E+04	4.51457143E+04	9.63421177E+03
8.00000000E+00	5.00000000E+04	8.41183787E+04	1.50683100E+04
8.00000000E+00	6.00000000E+04	1.41167232E+05	2.19496329E+04
8.00000000E+00	7.00000000E+04	2.18278806E+05	3.01231128E+04
8.00000000E+00	8.00000000E+04	3.15492982E+05	3.92557130E+04
8.00000000E+00	9.00000000E+04	4.30416864E+05	4.88468643E+04
8.00000000E+00	1.00000000E+05	5.58018816E+05	5.82628045E+04
1.20000000E+01	0.	-7.14752485E+02	4.79753089E+01
1.20000000E+01	1.00000000E+04	-7.97551179E+02	5.91713881E+02
1.20000000E+01	2.00000000E+04	1.10992291E+03	1.88839878E+03
1.20000000E+01	3.00000000E+04	7.48592875E+03	4.32166428E+03
1.20000000E+01	4.00000000E+04	2.15900656E+04	8.26236689E+03
1.20000000E+01	5.00000000E+04	4.70676864E+04	1.39983037E+04
1.20000000E+01	6.00000000E+04	8.73517251E+04	2.16636246E+04
1.20000000E+01	7.00000000E+04	1.44949811E+05	3.11801858E+04

1.20000000E+01	8.00000000E+04	2.20137612E+05	4.22222806E+04	
1.20000000E+01	9.00000000E+04	3.13390870E+05	5.42134565E+04	
1.20000000E+01	1.00000000E+05	4.19076310E+05	6.63598294E+04	
1.60000000E+01	0.	-5.01863233E+02	-2.83643164E+02	
1.60000000E+01	1.00000000E+04	-1.48038786E+03	-2.94360145E+02	
1.60000000E+01	2.00000000E+04	-2.14373506E+03	1.56524418E+02	
1.60000000E+01	3.00000000E+04	-8.91236506E+02	1.48655851E+03	
1.60000000E+01	4.00000000E+04	4.75239999E+03	4.16098818E+03	
1.60000000E+01	5.00000000E+04	1.78965130E+04	8.61192904E+03	
1.60000000E+01	6.00000000E+04	4.18079536E+04	1.51457286E+04	
1.60000000E+01	7.00000000E+04	7.92650575E+04	2.38546356E+04	
1.60000000E+01	8.00000000E+04	1.31853513E+05	3.45497539E+04	
1.60000000E+01	9.00000000E+04	1.99339526E+05	4.67301725E+04	
1.60000000E+01	1.00000000E+05	2.79253973E+05	5.95983440E+04	
2.00000000E+01	0.	6.08309797E+01	-2.41557542E+02	
2.00000000E+01	1.00000000E+04	-6.36318151E+02	-5.86991329E+02	
2.00000000E+01	2.00000000E+04	-1.82588813E+03	-8.48499095E+02	
2.00000000E+01	3.00000000E+04	-2.79018775E+03	-6.79007427E+02	
2.00000000E+01	4.00000000E+04	-2.01593220E+03	3.82911785E+02	
2.00000000E+01	5.00000000E+04	2.74663557E+03	2.84124228E+03	
2.00000000E+01	6.00000000E+04	1.41866286E+04	7.14377957E+03	
2.00000000E+01	7.00000000E+04	3.49270802E+04	1.35735029E+04	
2.00000000E+01	8.00000000E+04	6.68953752E+04	2.21509386E+04	
2.00000000E+01	9.00000000E+04	1.10692700E+05	3.25684783E+04	
2.00000000E+01	1.00000000E+05	1.65100610E+05	4.41737555E+04	
2.40000000E+01	0.	2.22244419E+02	5.81439048E+01	
2.40000000E+01	1.00000000E+04	1.90826511E+02	-1.94965024E+02	
2.40000000E+01	2.00000000E+04	-3.03403039E+02	-6.17760002E+02	
2.40000000E+01	3.00000000E+04	-1.18840698E+03	-1.02377214E+03	
2.40000000E+01	4.00000000E+04	-1.83821707E+03	-1.07328916E+03	
2.40000000E+01	5.00000000E+04	-9.74571706E+02	-3.14151754E+02	FLUTTER
2.40000000E+01	6.00000000E+04	3.23205619E+03	1.73709643E+03	
2.40000000E+01	7.00000000E+04	1.28466361E+04	5.48698493E+03	
2.40000000E+01	8.00000000E+04	2.96814460E+04	1.11506834E+04	
2.40000000E+01	9.00000000E+04	5.47306091E+04	1.86560189E+04	
2.40000000E+01	1.00000000E+05	8.76568555E+04	2.75896026E+04	
2.80000000E+01	0.	-1.49034195E+02	1.76655502E+02	
2.80000000E+01	1.00000000E+04	5.43066343E+00	2.43927180E+02	
2.80000000E+01	2.00000000E+04	1.02684455E+02	1.27601535E+02	
2.80000000E+01	3.00000000E+04	-3.12764635E+01	-1.66054056E+02	
2.80000000E+01	4.00000000E+04	-3.23981193E+02	-4.91181548E+02	
2.80000000E+01	5.00000000E+04	-2.78560047E+02	-5.58767000E+02	
2.80000000E+01	6.00000000E+04	1.07663911E+03	2.12405013E+01	
2.80000000E+01	7.00000000E+04	5.05599517E+03	1.65669934E+03	
2.80000000E+01	8.00000000E+04	1.29996066E+04	4.66463035E+03	
2.80000000E+01	9.00000000E+04	2.58286587E+04	9.16118626E+03	
2.80000000E+01	1.00000000E+05	4.35819049E+04	1.49777669E+04	
3.20000000E+01	0.	-3.96109287E+02	-9.71373469E+01	
3.20000000E+01	1.00000000E+04	-6.84970209E+02	1.18142987E+02	
3.20000000E+01	2.00000000E+04	-8.05485331E+02	3.18596078E+02	
3.20000000E+01	3.00000000E+04	-8.13604559E+02	4.04530416E+02	

3.20000000E+01	4.00000000E+04	-7.52520028E+02	3.51454508E+02	
3.20000000E+01	5.00000000E+04	-4.98960257E+02	2.52895420E+02	
3.20000000E+01	6.00000000E+04	3.43573681E+02	3.24532041E+02	
3.20000000E+01	7.00000000E+04	2.43079604E+03	8.62756790E+02	
3.20000000E+01	8.00000000E+04	6.50737357E+03	2.16383142E+03	
3.20000000E+01	9.00000000E+04	1.31024649E+04	4.42216088E+03	
3.20000000E+01	1.00000000E+05	2.21725059E+04	7.63473682E+03	
3.60000000E+01	0.	5.03155894E+01	-4.07285573E+02	
3.60000000E+01	1.00000000E+04	-6.84461920E+02	-4.11891110E+02	
3.60000000E+01	2.00000000E+04	-1.45734639E+03	-2.73279553E+02	
3.60000000E+01	3.00000000E+04	-2.07598879E+03	-6.84292707E+01	
3.60000000E+01	4.00000000E+04	-2.39251515E+03	1.19420861E+02	
3.60000000E+01	5.00000000E+04	-2.25332479E+03	2.57873902E+02	
3.60000000E+01	6.00000000E+04	-1.43901010E+03	4.02128330E+02	
3.60000000E+01	7.00000000E+04	3.41648663E+02	6.94474603E+02	
3.60000000E+01	8.00000000E+04	3.35962218E+03	1.32188871E+03	
3.60000000E+01	9.00000000E+04	7.67705981E+03	2.44137626E+03	
3.60000000E+01	1.00000000E+05	1.29187955E+04	4.09511989E+03	
4.00000000E+01	0.	7.39313282E+02	-2.05449836E+02	
4.00000000E+01	1.00000000E+04	2.16736263E+02	-5.59688366E+02	
4.00000000E+01	2.00000000E+04	-7.54123908E+02	-8.03987676E+02	
4.00000000E+01	3.00000000E+04	-1.88553473E+03	-8.98346482E+02	
4.00000000E+01	4.00000000E+04	-2.82052667E+03	-8.57130469E+02	
4.00000000E+01	5.00000000E+04	-3.19596339E+03	-7.11584212E+02	
4.00000000E+01	6.00000000E+04	-2.68666611E+03	-4.70395359E+02	
4.00000000E+01	7.00000000E+04	-1.06033155E+03	-9.92404701E+01	← FLUTTER
4.00000000E+01	8.00000000E+04	1.73106323E+03	4.68329372E+02	
4.00000000E+01	9.00000000E+04	5.42576759E+03	1.28941757E+03	
4.00000000E+01	1.00000000E+05	9.33476890E+03	2.35401225E+03	

The following printout is an optimization run with short printout option. The initial values of the flutter velocity and flutter frequency are taken from the previous computer run.

```
DETERMINATION OF THICKNESS DISTRIBUTION FOR MINIMUM WEIGHT WING
SATISFYING PRESCRIBED FLUTTER AND STRENGTH CRITERIA.
REFERENCE - NACA TN-3640,    STEIN-SANDERS,    JUNE,1956.
DEXTER-STROUD      RDK288       MAY,1969.


DATE 04/27/71


WING DESCRIPTION

    ASPECT RATIO............AR =   2.55380000E+00
    SURFACE AREA..........AREA =   5.65808000E+05 IN.**2
    L.E. SWEEP ANGLE.......ANG =   5.05000000E+01 DEGREES
    SEMISPAN................SS =   8.49988374E+02 IN.
    ROOT CHORD..............RC =   1.18122475E+03 IN.
    DEPTH RATIO.............DR =   3.00000000E-02
    WING MATERIAL.............     TITANIUM
    MASS DENSITY..........RHOW =   4.14000000E-04 (LB.*SEC.**2)/(IN.**4)
    YOUNGS MODULUS...........E =   1.64000000E+07 PSI
    POISSONS RATIO..........MU =   3.00000000E-01


AIR PROPERTIES

    SPEED OF SOUND...........A =   1.22040000E+04 IN./SEC. =   1.01700000E+03 FT./SEC.
    MASS DENSITY..........RHOA =   5.14000000E-08 (LB.*SEC.**2)/(IN.**4)
    CORRESPONDING ALTITUDE.... =   2.50000000E+04 FT.


FUEL PROPERTIES

    WEIGHT OF FUEL..........WTF =   1.50000000E+05 LB.
    VOLUME OF WING..........VOL =   9.03896471E+06 IN.**3
    DENSITY OF FUEL........RHOF =   4.29917638E-05 (LB.*SEC.**2)/(IN.**4)


EQUATION FOR THICKNESS - TBAR     (INCHES)

TBAR =    C( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3
       +(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B
       +(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*(1-B*B)
       +(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*(1-B*B)*B

        WHERE  A = 1-Y/SS    AND    B = 1-2*X/(LE+TE)

                          (ITHICK = 2)
```

```
                  XXX
                  XXX
                  XXX          FIXED BOUNDARY
                 .XXX          CONDITION
                . XXX
               .  XXX
              .   XXX
             .    XXX
            .     XXX
           .      XXX
          .       XXX  ROOT CHORD =1181.2 IN.
         .        XXX
        .         XXX
       .          XXX
      .           XXX
. 50 DEG.         XXX
. . . . . . . . .XXX
                  XXX
SEMISPAN =        XXX LEADING EDGE = A+B*Y
850.0 IN.         XXX         A = -1.18122475E+03
                              B =  1.21309700E+00
```

```
   DESIGN VARIABLES                 FINITE DIFFERENCE INCREMENTS     ANALYTICAL DERIVATIVES      * INDICATES ACTIVE
                                                                                                   DESIGN VARIABLES
   C( 1) =  2.00000000E-02          DC( 1) =  0.                     AD( 1) =  1.80836763E+05
   C( 2) =  0.                      DC( 2) =  0.                     AD( 2) =  1.13761448E+05
   C( 3) =  0.                      DC( 3) =  0.                     AD( 3) =  8.36219870E+04
   C( 4) =  0.                      DC( 4) =  0.                     AD( 4) =  6.62179501E+04
   C( 5) =  0.                      DC( 5) =  0.                     AD( 5) =  0.
   C( 6) =  0.                      DC( 6) =  0.                     AD( 6) =  0.
   C( 7) =  0.                      DC( 7) =  0.                     AD( 7) =  0.
   C( 8) =  0.                      DC( 8) =  0.                     AD( 8) =  0.
   C( 9) =  0.                      DC( 9) =  0.                     AD( 9) =  6.02789211E+04
 * C(10) =  2.30000000E-01          DC(10) =  1.00000000E-06         AD(10) =  7.58409650E+04
 * C(11) =  0.                      DC(11) =  1.00000000E-06         AD(11) =  5.57479913E+04
 * C(12) =  0.                      DC(12) =  1.00000000E-06         AD(12) =  4.41453001E+04
   C(13) =  0.                      DC(13) =  0.                     AD(13) =  0.
 * C(14) =  0.                      DC(14) =  1.00000000E-06         AD(14) =  0.
 * C(15) =  0.                      DC(15) =  1.00000000E-06         AD(15) =  0.
 * C(16) =  0.                      DC(16) =  1.00000000E-06         AD(16) =  0.

STATIONS

   NUMBER OF SEMISPAN STATIONS..................NN =  6
   NUMBER OF CHORDWISE STATIONS = (10 X NCS)...NCS =  1

INPUT OPTIONS

      IOPT = 2     1 - ITERATE ON OMEGA AND V, 2 - COMPLEX SEARCH
    IPRINT = 1     0 - MATRIX CONSTRUCTION, 1 - SHORT PRINTOUT, 2 - INTERMEDIATE PRINTOUT, 3 - EXTENDED PRINTOUT
   ISNDWCH = 1     0 - SOLID WING, 1 - SANDWICH CONSTRUCTION
       IHP = 1     0 - THICKNESS VARIATION EFFECTS NOT ACCOUNTED FOR IN PISTON THEORY, 1 - THICKNESS EFFECTS ARE USED


INITIAL WEIGHT =  2.10601572E+04   POUNDS


INPUT DATA FOR SEARCH ROUTINE

   NUMBER OF DESIGN VARIABLES....NC = 16
                                  R =  3.00000000E+02
   R REDUCTION FACTOR...........RDC =  1.50000000E+01
```

```
DESIGN CONSTRAINTS CONSIDERED
    FLUTTER - T
     STRESS - T
    MINGAGE - T


INPUT DATA FOR FLUTTER

    OMEGA INITIAL.............OMINIT =  2.40000000E+01 RADIANS/SEC.
    OMEGA INCREMENT............DELOM =  1.00000000E-04 RADIANS/SEC.

    V INITIAL..................VINIT =  5.00000000E+04 IN./SEC. =  4.16666667E+03 FT./SEC.
    V INCREMENT.................DELV =  1.00000000E-01 IN./SEC. =  8.33333333E-03 FT./SEC.

    V CRITICAL...................VCR =  2.50000000E+04 IN./SEC. =  2.08333333E+03 FT./SEC.


INPUT DATA FOR STRESS

    UNIFORM PRESSURE LOADING...PLOAD =  1.00000000E+00 PSI
    YIELD STRESS.................YCR =  1.25000000E+05 PSI


INPUT FOR MINGAGE

    MINIMUM GAGE.............TBARMIN=  2.00000000E-02 INCHES
```

OMEGA RADIANS/SEC.	V INCHES/SEC.	DETERMINANT REAL PART	DETERMINANT IMAGINARY PART
2.40000000E+01	5.00000000E+04	-9.74571706E+02	-3.14151754E+02
2.40001000E+01	5.00000000E+04	-9.74579281E+02	-3.14189575E+02
2.40000000E+01	5.00001000E+04	-9.74549739E+02	-3.14138482E+02
2.48261288E+01	5.47214022E+04	1.43529902E+02	5.25500697E+01
2.48262288E+01	5.47214022E+04	1.43503523E+02	5.25087797E+01
2.48261288E+01	5.47215022E+04	1.43561220E+02	5.25661784E+01
2.47493825E+01	5.41984633E+04	2.10816949E+00	7.31002324E-01
2.47494825E+01	5.41984633E+04	2.08451435E+00	6.90427856E-01
2.47493825E+01	5.41985633E+04	2.13828276E+00	7.46726934E-01
2.47480721E+01	5.41904331E+04	4.76938518E-04	1.78202736E-04
2.47481721E+01	5.41904331E+04	-2.31405444E-02	-4.03880499E-02
2.47480721E+01	5.41905331E+04	3.05730103E-02	1.58975004E-02
2.47480718E+01	5.41904313E+04	-5.58078517E-09	1.20520473E-09

```
COMPLEX DETERMINANT CONVERGED TO 0

FLUTTER Q =  1.08677798E+04   LB./SQ.FT.
```

STRESS TABLE

Y = 0.

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-6.78539E+02	-1.73257E+01	0.	-1.13250E-03	0.	-6.12296E+03	-2.04099E+04	0.	1.81407E+04
-8.46581E+02	-1.43903E+01	0.	-3.05253E-04	0.	-1.65038E+03	-5.50127E+03	0.	4.88963E+03
-9.91880E+02	-9.53962E+00	0.	-9.29141E-06	0.	-5.02349E+01	-1.67450E+02	0.	1.48832E+02
-1.10153E+03	-4.45910E+00	0.	1.46444E-05	0.	7.91761E+01	2.63920E+02	0.	2.34577E+02
-1.16581E+03	-9.12604E-01	0.	1.25482E-06	0.	6.78428E+00	2.26143E+01	0.	2.01000E+01
-5.02685E+02	-1.73257E+01	0.	-2.34589E-03	0.	-1.26833E+04	-4.22775E+04	0.	3.75771E+04
-3.34644E+02	-1.43903E+01	0.	-3.23914E-03	0.	-1.75127E+04	-5.83758E+04	0.	5.18855E+04
-1.89345E+02	-9.53962E+00	0.	-3.05825E-03	0.	-1.65347E+04	-5.51158E+04	0.	4.89880E+04
-7.96952E+01	-4.45910E+00	0.	-1.79997E-03	0.	-9.73171E+03	-3.24390E+04	0.	2.88325E+04
-1.54111E+01	-9.12604E-01	0.	-4.16854E-04	0.	-2.25376E+03	-7.51253E+03	0.	6.67728E+03

Y = 1.41664729E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-5.79821E+02	-1.48050E+01	-1.17071E-04	-1.53886E-03	-1.47589E-03	-1.04298E+04	-2.83662E+04	-9.30947E+03	2.96238E+04
-7.23414E+02	-1.22967E+01	-9.72362E-05	-7.64465E-04	-1.00773E-03	-5.88554E+03	-1.43029E+04	-6.35644E+03	1.66206E+04
-8.47574E+02	-8.15173E+00	-6.44598E-05	-2.23712E-04	-5.43021E-04	-2.37121E+03	-4.38024E+03	-3.42521E+03	7.04406E+03
-9.41271E+02	-3.81036E+00	-3.01304E-05	-7.98297E-06	-2.09723E-04	-5.86170E+02	-3.06772E+02	-1.32287E+03	2.34688E+03
-9.96203E+02	-7.79832E-01	-6.16652E-06	9.68813E-06	-3.76306E-05	-5.87531E+01	1.41259E+02	-2.37362E+02	4.48027E+02
-4.29551E+02	-1.48050E+01	-1.17071E-04	-2.21350E-03	-1.75070E-03	-1.40773E+04	-4.05246E+04	-1.10429E+04	4.04447E+04
-2.85957E+02	-1.22967E+01	-9.72362E-05	-2.39570E-03	-1.67221E-03	-1.47050E+04	-4.37010E+04	-1.05478E+04	4.26281E+04
-1.61797E+02	-8.15173E+00	-6.44598E-05	-1.91893E-03	-1.23356E-03	-1.15366E+04	-3.49314E+04	-7.78093E+03	3.36442E+04
-6.81006E+01	-3.81036E+00	-3.01304E-05	-1.01690E-03	-6.20704E-04	-6.04100E+03	-1.84895E+04	-3.91521E+03	1.76819E+04
-1.31690E+01	-7.79832E-01	-6.16652E-06	-2.22779E-04	-1.32325E-04	-1.31561E+03	-4.04826E+03	-8.34668E+02	3.85789E+03

Y = 2.83329458E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-4.81102E+02	-1.22843E+01	-2.14966E-04	-1.62507E-03	-2.37007E-03	-1.26602E+04	-3.04492E+04	-1.49497E+04	3.70462E+04
-6.00248E+02	-1.02031E+01	-1.78546E-04	-1.01379E-03	-1.83123E-03	-8.69888E+03	-1.92357E+04	-1.15508E+04	2.60503E+04
-7.03268E+02	-6.76383E+00	-1.18362E-04	-4.67209E-04	-1.13525E-03	-4.65912E+03	-9.05996E+03	-7.16084E+03	1.46769E+04
-7.81013E+02	-3.16162E+00	-5.53258E-05	-1.42604E-04	-5.02890E-04	-1.76808E+03	-2.86913E+03	-3.17208E+03	6.03915E+03
-8.26592E+02	-6.47059E-01	-1.13230E-05	-1.98040E-05	-9.95910E-05	-3.11136E+02	-4.18126E+02	-6.28189E+02	1.15126E+03
-3.56417E+02	-1.22843E+01	-2.14966E-04	-2.01875E-03	-2.54307E-03	-1.47887E+04	-3.75440E+04	-1.60409E+04	4.29530E+04
-2.37271E+02	-1.02031E+01	-1.78546E-04	-1.96567E-03	-2.24951E-03	-1.38453E+04	-3.63905E+04	-1.41892E+04	4.02011E+04
-1.34250E+02	-6.76383E+00	-1.18362E-04	-1.45643E-03	-1.56995E-03	-1.00074E+04	-2.68876E+04	-9.90275E+03	2.91240E+04
-5.65060E+01	-3.16162E+00	-5.53258E-05	-7.31344E-04	-7.61602E-04	-4.95116E+03	-1.34794E+04	-4.80395E+03	1.44466E+04
-1.09269E+01	-6.47059E-01	-1.13230E-05	-1.55457E-04	-1.59201E-04	-1.04455E+03	-2.86286E+03	-1.00419E+03	3.05317E+03

Y = 4.24994187E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-3.82383E+02	-9.76368E+00	-2.33427E-04	-1.57290E-03	-2.71956E-03	-1.27109E+04	-2.96088E+04	-1.71542E+04	3.93025E+04
-4.77081E+02	-8.10949E+00	-1.93879E-04	-1.09076E-03	-2.20427E-03	-9.39139E+03	-2.07059E+04	-1.39038E+04	3.00404E+04
-5.58963E+02	-5.37594E+00	-1.28526E-04	-5.94136E-04	-1.42999E-03	-5.52855E+03	-1.14024E+04	-9.01994E+03	1.84829E+04
-6.20754E+02	-2.51287E+00	-6.00771E-05	-2.30698E-04	-6.57393E-04	-2.33000E+03	-4.48245E+03	-4.14663E+03	8.16461E+03
-6.56981E+02	-5.14287E-01	-1.22954E-05	-4.14479E-05	-1.33219E-04	-4.45680E+02	-8.13450E+02	-8.40307E+02	1.61745E+03
-2.83283E+02	-9.76368E+00	-2.33427E-04	-1.83174E-03	-2.78828E-03	-1.41103E+04	-3.42736E+04	-1.75876E+04	4.26396E+04
-1.88585E+02	-8.10949E+00	-1.93879E-04	-1.71661E-03	-2.37042E-03	-1.27751E+04	-3.19850E+04	-1.49519E+04	3.80556E+04
-1.06703E+02	-5.37594E+00	-1.28526E-04	-1.24454E-03	-1.60266E-03	-9.04501E+03	-2.31239E+04	-1.01091E+04	2.67199E+04
-4.49113E+01	-2.51287E+00	-6.00771E-05	-6.17790E-04	-7.60157E-04	-4.42285E+03	-1.14586E+04	-4.79484E+03	1.30059E+04
-8.68476E+00	-5.14287E-01	-1.22954E-05	-1.30638E-04	-1.56898E-04	-9.27897E+02	-2.42084E+03	-9.89662E+02	2.72277E+03

Y = 5.66658916E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-2.83664E+02	-7.24302E+00	-1.99618E-04	-1.38825E-03	-2.68001E-03	-1.11032E+04	-2.60983E+04	-1.69047E+04	3.70396E+04
-3.53914E+02	-6.01588E+00	-1.65798E-04	-1.01108E-03	-2.20495E-03	-8.45450E+03	-1.91181E+04	-1.39082E+04	2.92516E+04
-4.14657E+02	-3.98805E+00	-1.09911E-04	-5.88679E-04	-1.44967E-03	-5.16356E+03	-1.12034E+04	-9.14406E+03	1.85788E+04
-4.60496E+02	-1.86413E+00	-5.13755E-05	-2.46327E-04	-6.73371E-04	-2.25768E+03	-4.71707E+03	-4.24742E+03	8.41546E+03
-4.87370E+02	-3.81515E-01	-1.05146E-05	-4.69484E-05	-1.37303E-04	-4.43324E+02	-9.02951E+02	-8.65067E+02	1.69168E+03
-2.10148E+02	-7.24302E+00	-1.99618E-04	-1.56662E-03	-2.70648E-03	-1.20676E+04	-2.93129E+04	-1.70716E+04	3.90571E+04
-1.39898E+02	-6.01588E+00	-1.65798E-04	-1.44237E-03	-2.26894E-03	-1.07863E+04	-2.68908E+04	-1.43118E+04	3.41158E+04
-7.91558E+01	-3.98805E+00	-1.09911E-04	-1.03689E-03	-1.51616E-03	-7.58683E+03	-1.92810E+04	-9.56350E+03	2.36097E+04
-3.33167E+01	-1.86413E+00	-5.13755E-05	-5.13081E-04	-7.12947E-04	-3.69991E+03	-9.52451E+03	-4.49705E+03	1.13946E+04
-6.44264E+00	-3.81515E-01	-1.05146E-05	-1.08412E-04	-1.46422E-04	-7.75631E+02	-2.01064E+03	-9.23585E+02	2.37567E+03

Y = 7.08323645E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-1.84945E+02	-4.72236E+00	-1.46148E-04	-8.37962E-04	-2.42900E-03	-7.16440E+03	-1.58919E+04	-1.53214E+04	2.99041E+04
-2.30748E+02	-3.92228E+00	-1.21387E-04	-5.68480E-04	-2.07516E-03	-5.26119E+03	-1.09014E+04	-1.30895E+04	2.45595E+04
-2.70351E+02	-2.60016E+00	-8.04702E-05	-2.92315E-04	-1.40873E-03	-3.03066E+03	-5.70316E+03	-8.88584E+03	1.61648E+04
-3.00237E+02	-1.21539E+00	-3.76141E-05	-1.03528E-04	-6.70147E-04	-1.23761E+03	-2.06914E+03	-4.22708E+03	7.54035E+03
-3.17759E+02	-2.48743E-01	-7.69815E-06	-1.69464E-05	-1.38553E-04	-2.30358E+02	-3.47028E+02	-8.73947E+02	1.54431E+03
-1.37014E+02	-4.72236E+00	-1.46148E-04	-9.70993E-04	-2.35631E-03	-7.88364E+03	-1.82894E+04	-1.48629E+04	3.02520E+04
-9.12119E+01	-3.92228E+00	-1.21387E-04	-8.90139E-04	-1.89941E-03	-7.00026E+03	-1.66984E+04	-1.19809E+04	2.53292E+04
-5.16086E+01	-2.60016E+00	-8.04702E-05	-6.26589E-04	-1.22609E-03	-4.83795E+03	-1.17274E+04	-7.73377E+03	1.68415E+04
-2.17221E+01	-1.21539E+00	-3.76141E-05	-3.02474E-04	-5.61445E-04	-2.31324E+03	-5.65455E+03	-3.54142E+03	7.86572E+03
-4.20052E+00	-2.48743E-01	-7.69815E-06	-6.27859E-05	-1.13506E-04	-4.78194E+02	-1.17315E+03	-7.15962E+02	1.60679E+03

Y = 8.49958374E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-8.62267E+01	-2.20169E+00	3.94811E-05	-1.18443E-05	-1.18526E-03	6.47491E+02	3.26299E-08	-7.47623E+03	1.29654E+04
-1.07581E+02	-1.82868E+00	3.27921E-05	-9.83764E-06	-1.01139E-03	5.37791E+02	1.06070E-08	-6.37956E+03	1.10628E+04
-1.26045E+02	-1.21227E+00	2.17386E-05	-6.52157E-06	-6.85920E-04	3.56512E+02	-1.54425E-08	-4.32657E+03	7.50231E+03
-1.39979E+02	-5.66649E-01	1.01612E-05	-3.04837E-06	-3.26068E-04	1.66644E+02	-1.88827E-08	-2.05674E+03	3.56627E+03
-1.48148E+02	-1.15971E-01	2.07961E-06	-6.23883E-07	-6.73871E-05	3.41056E+01	-5.56997E-09	-4.25057E+02	7.37010E+02
-6.38797E+01	-2.20169E+00	3.94811E-05	-1.18443E-05	-1.15130E-03	6.47491E+02	2.20024E-08	-7.26206E+03	1.25949E+04
-4.25255E+01	-1.82868E+00	3.27921E-05	-9.83764E-06	-9.29298E-04	5.37791E+02	-1.50903E-08	-5.86172E+03	1.01670E+04
-2.40613E+01	-1.21227E+00	2.17386E-05	-6.52157E-06	-6.00603E-04	3.56512E+02	-4.21461E-08	-3.78842E+03	6.57142E+03
-1.01274E+01	-5.66649E-01	1.01612E-05	-3.04837E-06	-2.75291E-04	1.66644E+02	-3.47754E-08	-1.73645E+03	3.01224E+03
-1.95840E+00	-1.15971E-01	2.07961E-06	-6.23883E-07	-5.56876E-05	3.41056E+01	-9.23191E-09	-3.51260E+02	6.09356E+02

```
FAILURE RATIO 1 =  4.61336059E-01     G1 =  5.38663941E-01
FAILURE RATIO 2 =  2.17128178E-01     G2 =  7.82871822E-01
FAILURE RATIO 3 =  3.23765185E-01     G3 =  6.76234815E-01


   WT = WEIGHT IN POUNDS =  2.10601572E+04
   PF = PENALTY FUNCTION =  2.53620548E+04
PFUNC =      WT + PF     =  4.64222120E+04

CONSTANTS 1 - 16
     2.00000000E-02      0.                  0.                  0.
     0.                  0.                  0.                  0.
     0.                  2.30000000E-01      0.                  0.
     0.                  0.                  0.                  0.

***********************************************************************************
```

THE FOLLOWING INFORMATION IS FOR THE FINAL DESIGN

CTEST = 7.77971542E-03

ESTIMATED MINIMUM WEIGHT LESS THAN 2 PERCENT LOWER THAN PRESENT WEIGHT
DESIGN CONSIDERED CONVERGED

FINAL WEIGHT= 8.13621051E+03

FINAL CONSTANTS

2.00000000E-02	0.	0.	0.
0.	0.	0.	0.
0.	3.43637754E-03	1.49277271E-01	-9.20381315E-02
0.	7.56679020E-02	-4.68304085E-01	4.50574029E-01

OMEGA RADIANS/SEC.	V INCHES/SEC.	DETERMINANT REAL PART	DETERMINANT IMAGINARY PART
1.59740137E+01	2.50952961E+04	9.45799314E-17	7.84263020E-17
1.59741137E+01	2.50952961E+04	-1.72088298E-09	-1.15412245E-08
1.59740137E+01	2.50953961E+04	5.56490046E-09	5.55167102E-09
1.59740137E+01	2.50952961E+04	-9.11912192E-17	4.76074509E-17

COMPLEX DETERMINANT CONVERGED TO 0

FLUTTER Q = 2.33066720E+03 LB./SQ.FT.

STRESS TABLE

Y = 0.

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-6.78539E+02	-1.73257E+01	0.	-3.53132E-03	0.	-1.90924E+04	-6.36414E+04	0.	5.65658E+04
-8.46581E+02	-1.43903E+01	0.	-1.30425E-03	0.	-7.05154E+03	-2.35051E+04	0.	2.08918E+04
-9.91880E+02	-9.53962E+00	0.	-1.87883E-04	0.	-1.01581E+03	-3.38602E+03	0.	3.00956E+03
-1.10153E+03	-4.45910E+00	0.	7.71594E-05	0.	4.17169E+02	1.39056E+03	0.	1.23596E+03
-1.16581E+03	-9.12604E-01	0.	2.95516E-05	0.	1.59773E+02	5.32578E+02	0.	4.73365E+02
-5.02685E+02	-1.73257E+01	0.	-6.20327E-03	0.	-3.35386E+04	-1.11795E+05	0.	9.93658E+04
-3.34644E+02	-1.43903E+01	0.	-7.76484E-03	0.	-4.19813E+04	-1.39938E+05	0.	1.24379E+05
-1.89345E+02	-9.53962E+00	0.	-6.90187E-03	0.	-3.73156E+04	-1.24385E+05	0.	1.10556E+05
-7.96952E+01	-4.45910E+00	0.	-3.91873E-03	0.	-2.11870E+04	-7.06232E+04	0.	6.27713E+04
-1.54111E+01	-9.12604E-01	0.	-8.91147E-04	0.	-4.81807E+03	-1.60602E+04	0.	1.42746E+04

Y = 1.41664729E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-5.79821E+02	-1.48050E+01	-1.75742E-04	-4.27863E-03	-3.56247E-03	-2.63000E+04	-7.80596E+04	-2.24710E+04	7.90370E+04
-7.23414E+02	-1.22967E+01	-1.45967E-04	-2.21393E-03	-2.48865E-03	-1.46004E+04	-4.06886E+04	-1.56976E+04	4.48766E+04
-8.47574E+02	-8.15173E+00	-9.67646E-05	-7.94124E-04	-1.38023E-03	-6.03739E+03	-1.48348E+04	-8.70604E+03	1.98581E+04
-9.41271E+02	-3.81036E+00	-4.52306E-05	-1.60717E-04	-5.50075E-04	-1.68408E+03	-3.14099E+03	-3.46971E+03	6.59764E+03
-9.96203E+02	-7.79832E-01	-9.25695E-06	-9.85667E-06	-1.01170E-04	-2.20119E+02	-2.27685E+02	-6.38151E+02	1.12778E+03
-4.29551E+02	-1.48050E+01	-1.75742E-04	-6.19456E-03	-4.15498E-03	-3.66587E+04	-1.12588E+05	-2.62083E+04	1.09331E+05
-2.85957E+02	-1.22967E+01	-1.45967E-04	-6.84652E-03	-3.92128E-03	-3.96470E+04	-1.24177E+05	-2.47343E+04	1.17915E+05
-1.61797E+02	-8.15173E+00	-9.67646E-05	-5.60842E-03	-2.86905E-03	-3.20663E+04	-1.01598E+05	-1.80971E+04	9.52630E+04
-6.81006E+01	-3.81036E+00	-4.52306E-05	-3.02598E-03	-1.43616E-03	-1.71754E+04	-5.47788E+04	-9.05887E+03	5.10005E+04
-1.31690E+01	-7.79832E-01	-9.25695E-06	-6.70047E-04	-3.05335E-04	-3.78950E+03	-1.21256E+04	-1.92596E+03	1.12503E+04

Y = 2.83329458E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-4.81102E+02	-1.22843E+01	-4.63473E-04	-4.44427E-03	-6.25005E-03	-3.23811E+04	-8.26004E+04	-3.94234E+04	9.92939E+04
-6.00248E+02	-1.02031E+01	-3.84950E-04	-2.71632E-03	-4.66743E-03	-2.16236E+04	-5.10348E+04	-2.94407E+04	6.75932E+04
-7.03268E+02	-6.76383E+00	-2.55191E-04	-1.27777E-03	-2.79393E-03	-1.15074E+04	-2.44077E+04	-1.76233E+04	3.71352E+04
-7.81013E+02	-3.16162E+00	-1.19284E-04	-4.23116E-04	-1.20007E-03	-4.43735E+03	-8.27031E+03	-7.56969E+03	1.49429E+04
-8.26592E+02	-6.47059E-01	-2.44128E-05	-6.65436E-05	-2.32902E-04	-7.99741E+02	-1.33124E+03	-1.46908E+03	2.79673E+03
-3.56417E+02	-1.22843E+01	-4.63473E-04	-5.75938E-03	-6.90992E-03	-3.94913E+04	-1.06301E+05	-4.35857E+04	1.19836E+05
-2.37271E+02	-1.02031E+01	-3.84950E-04	-5.89617E-03	-6.26295E-03	-3.88157E+04	-1.08342E+05	-3.95047E+04	1.17137E+05
-1.34250E+02	-6.76383E+00	-2.55191E-04	-4.58233E-03	-4.45203E-03	-2.93739E+04	-8.39624E+04	-2.80820E+04	8.83856E+04
-5.65060E+01	-3.16162E+00	-1.19284E-04	-2.38985E-03	-2.18690E-03	-1.50707E+04	-4.37148E+04	-1.37943E+04	4.52786E+04
-1.09269E+01	-6.47059E-01	-2.44128E-05	-5.19703E-04	-4.60280E-04	-3.24979E+03	-9.49806E+03	-2.90330E+03	9.75680E+03

Y = 4.24994187E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-3.82383E+02	-9.76368E+00	-7.11790E-04	-4.34126E-03	-7.79762E-03	-3.62993E+04	-8.20864E+04	-4.91850E+04	1.11057E+05
-4.77081E+02	-8.10949E+00	-5.91196E-04	-2.92056E-03	-6.14547E-03	-2.64448E+04	-5.58306E+04	-3.87637E+04	8.27516E+04
-5.58963E+02	-5.37594E+00	-3.91916E-04	-1.52538E-03	-3.88420E-03	-1.53102E+04	-2.96093E+04	-2.45003E+04	4.95841E+04
-6.20754E+02	-2.51287E+00	-1.83193E-04	-5.62974E-04	-1.74865E-03	-6.34527E+03	-1.11363E+04	-1.10300E+04	2.14149E+04
-6.56981E+02	-5.14287E-01	-3.74925E-05	-9.67946E-05	-3.49849E-04	-1.19902E+03	-1.94714E+03	-2.20674E+03	4.18371E+03
-2.83283E+02	-9.76368E+00	-7.11790E-04	-5.16123E-03	-8.21473E-03	-4.07325E+04	-9.68639E+04	-5.18160E+04	1.23089E+05
-1.88585E+02	-8.10949E+00	-5.91196E-04	-4.90319E-03	-7.15401E-03	-3.71641E+04	-9.15615E+04	-4.51253E+04	1.11673E+05
-1.06703E+02	-5.37594E+00	-3.91916E-04	-3.58578E-03	-4.93230E-03	-2.64499E+04	-6.67417E+04	-3.11114E+04	7.93253E+04
-4.49113E+01	-2.51287E+00	-1.83193E-04	-1.78924E-03	-2.37244E-03	-1.29752E+04	-3.32360E+04	-1.49646E+04	3.89045E+04
-8.68476E+00	-5.14287E-01	-3.74925E-05	-3.79340E-04	-4.93577E-04	-2.72662E+03	-7.03916E+03	-3.11333E+03	8.17738E+03

Y = 5.66658916E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-2.83664E+02	-7.24302E+00	-7.15596E-04	-3.79068E-03	-7.78350E-03	-3.33911E+04	-7.21845E+04	-4.90959E+04	1.05577E+05
-3.53914E+02	-6.01588E+00	-5.94358E-04	-2.75454E-03	-6.49611E-03	-2.56042E+04	-5.28558E+04	-4.09754E+04	8.44567E+04
-4.14657E+02	-3.98805E+00	-3.94011E-04	-1.55101E-03	-4.32434E-03	-1.54865E+04	-3.00825E+04	-2.72766E+04	5.39533E+04
-4.60496E+02	-1.86413E+00	-1.84173E-04	-6.13838E-04	-2.02766E-03	-6.63793E+03	-1.20583E+04	-1.27898E+04	2.44982E+04
-4.87370E+02	-3.81515E-01	-3.76929E-05	-1.11135E-04	-4.15742E-04	-1.28017E+03	-2.20667E+03	-2.62237E+03	4.93090E+03
-2.10148E+02	-7.24302E+00	-7.15596E-04	-4.16819E-03	-7.74405E-03	-3.54322E+04	-7.89880E+04	-4.88471E+04	1.08876E+05
-1.39898E+02	-6.01588E+00	-5.94358E-04	-3.66734E-03	-6.40073E-03	-3.05393E+04	-6.93062E+04	-4.03738E+04	9.22473E+04
-7.91558E+01	-3.98805E+00	-3.94011E-04	-2.49961E-03	-4.22522E-03	-2.06152E+04	-4.71782E+04	-2.66514E+04	6.17177E+04
-3.33167E+01	-1.86413E+00	-1.84173E-04	-1.17841E-03	-1.96866E-03	-9.69031E+03	-2.22329E+04	-1.24177E+04	2.89026E+04
-6.44264E+00	-3.81515E-01	-3.76929E-05	-2.41218E-04	-4.02149E-04	-1.98347E+03	-4.55102E+03	-2.53663E+03	5.90954E+03

Y = 7.08323645E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-1.84945E+02	-4.72236E+00	-2.94707E-04	-2.03177E-03	-5.68909E-03	-1.62962E+04	-3.82099E+04	-3.58850E+04	7.04705E+04
-2.30748E+02	-3.92228E+00	-2.44777E-04	-1.56758E-03	-5.05592E-03	-1.28866E+04	-2.95742E+04	-3.18912E+04	6.09158E+04
-2.70351E+02	-2.60016E+00	-1.62267E-04	-9.29221E-04	-3.54122E-03	-7.94830E+03	-1.76237E+04	-2.23369E+04	4.15994E+04
-3.00237E+02	-1.21539E+00	-7.58486E-05	-3.83737E-04	-1.72214E-03	-3.44165E+03	-7.32578E+03	-1.08627E+04	1.98569E+04
-3.17759E+02	-2.48743E-01	-1.55233E-05	-7.14958E-05	-3.60477E-04	-6.66308E+02	-1.37242E+03	-2.27378E+03	4.11379E+03
-1.37014E+02	-4.72236E+00	-2.94707E-04	-2.08356E-03	-5.27243E-03	-1.65761E+04	-3.91432E+04	-3.32568E+04	6.69042E+04
-9.12119E+01	-3.92228E+00	-2.44777E-04	-1.69278E-03	-4.04846E-03	-1.35636E+04	-3.18307E+04	-2.55364E+04	5.21705E+04
-5.16086E+01	-2.60016E+00	-1.62267E-04	-1.05934E-03	-2.49425E-03	-8.65180E+03	-1.99687E+04	-1.57330E+04	3.23020E+04
-2.17221E+01	-1.21539E+00	-7.58486E-05	-4.61179E-04	-1.09902E-03	-3.86035E+03	-8.72143E+03	-6.93230E+03	1.41940E+04
-4.20052E+00	-2.48743E-01	-1.55233E-05	-8.93393E-05	-2.16904E-04	-7.62781E+02	-1.69400E+03	-1.36817E+03	2.78836E+03

Y = 8.49988374E+02

X	-ZU	EP X	EP Y	GAM XY	SIG X	SIG Y	TAU XY	Y
-8.62267E+01	-2.20169E+00	3.56621E-04	-1.06986E-04	-2.10527E-03	5.84858E+03	-2.33458E-08	-1.32794E+04	2.37325E+04
-1.07581E+02	-1.82868E+00	2.96201E-04	-8.88603E-05	-1.87229E-03	4.85770E+03	-4.24397E-09	-1.18098E+04	2.10241E+04
-1.26045E+02	-1.21227E+00	1.96358E-04	-5.89073E-05	-1.31208E-03	3.22027E+03	1.23724E-08	-8.27623E+03	1.46921E+04
-1.39979E+02	-5.66649E-01	9.17834E-05	-2.75350E-05	-6.38319E-04	1.50525E+03	1.30074E-08	-4.02632E+03	7.13439E+03
-1.48148E+02	-1.15971E-01	1.87845E-05	-5.63534E-06	-1.33640E-04	3.08065E+02	3.68221E-09	-8.42961E+02	1.49220E+03
-6.38797E+01	-2.20169E+00	3.56621E-04	-1.06986E-04	-1.94941E-03	5.84858E+03	-2.67300E-08	-1.22963E+04	2.20862E+04
-4.25255E+01	-1.82868E+00	2.96201E-04	-8.88603E-05	-1.49543E-03	4.85770E+03	-1.24193E-08	-9.43273E+03	1.70448E+04
-2.40613E+01	-1.21227E+00	1.96358E-04	-5.89073E-05	-9.20446E-04	3.22027E+03	3.88054E-09	-5.80589E+03	1.05591E+04
-1.01274E+01	-5.66649E-01	9.17834E-05	-2.75350E-05	-4.05232E-04	1.50525E+03	7.95452E-09	-2.55608E+03	4.67615E+03
-1.95840E+00	-1.15971E-01	1.87845E-05	-5.63534E-06	-7.99341E-05	3.08065E+02	2.51815E-09	-5.04200E+02	9.26043E+02

FAILURE RATIO 1 = 9.96202630E-01 G1 = 3.75736952E-03
FAILURE RATIO 2 = 9.00804250E-01 G2 = 9.91957504E-02
FAILURE RATIO 3 = 8.02510251E-01 G3 = 1.97489749E-01

THICKNESS TABLE

Y	X	TEAR
0	-1.18122475E+03	2.00000000E-02
0	-1.12216351E+03	2.16209764E-02
0	-1.06310228E+03	2.51570862E-02
0	-1.00404104E+03	3.02607023E-02
0	-9.44979800E+02	3.65841976E-02
0	-8.85918563E+02	4.37799450E-02
0	-8.26857325E+02	5.15003176E-02
0	-7.67796088E+02	5.93976881E-02
0	-7.08734850E+02	6.71244295E-02
0	-6.49673613E+02	7.43329148E-02
0	-5.90612375E+02	8.06755168E-02
0	-5.31551138E+02	8.58046084E-02
0	-4.72489900E+02	8.93725627E-02
0	-4.13428663E+02	9.10317524E-02
0	-3.54367425E+02	9.04345506E-02
0	-2.95306188E+02	8.72333301E-02
0	-2.36244950E+02	8.10804639E-02
0	-1.77183713E+02	7.16283249E-02
0	-1.18122475E+02	5.85292859E-02
0	-5.90612375E+01	4.14357200E-02
0	3.91082722E-11	2.00000000E-02
71	-1.09529822E+03	2.00000000E-02
71	-1.04053331E+03	2.70432003E-02
71	-9.85768399E+02	3.41698956E-02
71	-9.31003488E+02	4.12426052E-02
71	-8.76238577E+02	4.81238484E-02
71	-8.21473666E+02	5.46761444E-02
71	-7.66708755E+02	6.07620123E-02
71	-7.11943844E+02	6.62439715E-02
71	-6.57178933E+02	7.09845413E-02
71	-6.02414022E+02	7.48462407E-02
71	-5.47649111E+02	7.76915892E-02
71	-4.92884200E+02	7.93831058E-02
71	-4.38119289E+02	7.97833099E-02
71	-3.83354377E+02	7.87547208E-02
71	-3.28589466E+02	7.61598575E-02
71	-2.73824555E+02	7.18612394E-02
71	-2.19059644E+02	6.57213857E-02
71	-1.64294733E+02	5.76028157E-02
71	-1.09529822E+02	4.73680486E-02
71	-5.47649111E+01	3.48796036E-02
71	4.43378667E-11	2.00000000E-02

142	-1.00937169E+03	2.00000000E-02
142	-9.58903108E+02	3.03608452E-02
142	-9.08434523E+02	3.95804691E-02
142	-8.57965938E+02	4.76673059E-02
142	-8.07497354E+02	5.46297900E-02
142	-7.57028769E+02	6.04763558E-02
142	-7.06560185E+02	6.52154375E-02
142	-6.56091600E+02	6.88554697E-02
142	-6.05623015E+02	7.14048865E-02
142	-5.55154431E+02	7.28721224E-02
142	-5.04685846E+02	7.32656117E-02
142	-4.54217261E+02	7.25937888E-02
142	-4.03748677E+02	7.08650880E-02
142	-3.53280092E+02	6.80879437E-02
142	-3.02811508E+02	6.42707901E-02
142	-2.52342923E+02	5.94220618E-02
142	-2.01874338E+02	5.35501930E-02
142	-1.51405754E+02	4.66636181E-02
142	-1.00937169E+02	3.87707713E-02
142	-5.04685846E+01	2.98800872E-02
142	6.36646291E-12	2.00000000E-02
212	-9.23445163E+02	2.00000000E-02
212	-8.77272905E+02	3.19021592E-02
212	-8.31100647E+02	4.19544283E-02
212	-7.84928389E+02	5.02563125E-02
212	-7.38756130E+02	5.69073169E-02
212	-6.92583872E+02	6.20069468E-02
212	-6.46411614E+02	6.56547074E-02
212	-6.00239356E+02	6.79501040E-02
212	-5.54067098E+02	6.89926416E-02
212	-5.07894840E+02	6.88818256E-02
212	-4.61722582E+02	6.77171612E-02
212	-4.15550323E+02	6.55981536E-02
212	-3.69378065E+02	6.26243079E-02
212	-3.23205807E+02	5.88951294E-02
212	-2.77033549E+02	5.45101234E-02
212	-2.30861291E+02	4.95687950E-02
212	-1.84689033E+02	4.41706494E-02
212	-1.38516774E+02	3.84151919E-02
212	-9.23445163E+01	3.24019277E-02
212	-4.61722582E+01	2.62303620E-02
212	1.70530257E-11	2.00000000E-02
283	-8.37518634E+02	2.00000000E-02
283	-7.95642702E+02	3.19953902E-02
283	-7.53766771E+02	4.18573950E-02
283	-7.11890839E+02	4.97311333E-02
283	-6.70014907E+02	5.57617237E-02
283	-6.28138975E+02	6.00942851E-02
283	-5.86263044E+02	6.28739363E-02
283	-5.44387112E+02	6.42457961E-02
283	-5.02511180E+02	6.43549832E-02

283	-4.60635249E+02	6.33466164E-02
283	-4.18759317E+02	6.13658145E-02
283	-3.76883385E+02	5.85576964E-02
283	-3.35007454E+02	5.50673807E-02
283	-2.93131522E+02	5.10399864E-02
283	-2.51255590E+02	4.66206321E-02
283	-2.09379658E+02	4.19544367E-02
283	-1.67503727E+02	3.71865189E-02
283	-1.25627795E+02	3.24619976E-02
283	-8.37518634E+01	2.79259914E-02
283	-4.18759317E+01	2.37236193E-02
283	1.45519152E-11	2.00000000E-02
354	-7.51592105E+02	2.00000000E-02
354	-7.14012500E+02	3.09687861E-02
354	-6.76432894E+02	3.98549909E-02
354	-6.38853289E+02	4.68132765E-02
354	-6.01273684E+02	5.19983049E-02
354	-5.63694079E+02	5.55647382E-02
354	-5.26114473E+02	5.76672384E-02
354	-4.88534868E+02	5.84604675E-02
354	-4.50955263E+02	5.80990875E-02
354	-4.13375658E+02	5.67377605E-02
354	-3.75796052E+02	5.45311485E-02
354	-3.38216447E+02	5.16339135E-02
354	-3.00636842E+02	4.82007176E-02
354	-2.63057237E+02	4.43862228E-02
354	-2.25477631E+02	4.03450911E-02
354	-1.87898026E+02	3.62319846E-02
354	-1.50318421E+02	3.22015652E-02
354	-1.12738816E+02	2.84084950E-02
354	-7.51592105E+01	2.50074361E-02
354	-3.75796052E+01	2.21530504E-02
354	1.86446414E-11	2.00000000E-02
425	-6.65665576E+02	2.00000000E-02
425	-6.32382297E+02	2.91505947E-02
425	-5.99099018E+02	3.65128375E-02
425	-5.65815739E+02	4.22242504E-02
425	-5.32532461E+02	4.64223551E-02
425	-4.99249182E+02	4.92446737E-02
425	-4.65965903E+02	5.08287279E-02
425	-4.32682624E+02	5.13120398E-02
425	-3.99399345E+02	5.08321312E-02
425	-3.66116067E+02	4.95265239E-02
425	-3.32832788E+02	4.75327400E-02
425	-2.99549509E+02	4.49883013E-02
425	-2.66266230E+02	4.20307296E-02
425	-2.32982951E+02	3.87975470E-02
425	-1.99699673E+02	3.54262753E-02
425	-1.66416394E+02	3.20544363E-02
425	-1.33133115E+02	2.88195521E-02
425	-9.98498363E+01	2.58591444E-02

425	-6.65665576E+01	2.33107353E-02
425	-3.32832788E+01	2.13118465E-02
425	2.43289833E-11	2.00000000E-02
496	-5.79739047E+02	2.00000000E-02
496	-5.50752094E+02	2.68690641E-02
496	-5.21765142E+02	3.23965568E-02
496	-4.92778190E+02	3.66855632E-02
496	-4.63791237E+02	3.98391689E-02
496	-4.34804285E+02	4.19604591E-02
496	-4.05817333E+02	4.31525192E-02
496	-3.76830380E+02	4.35184346E-02
496	-3.47843428E+02	4.31612906E-02
496	-3.18856476E+02	4.21841726E-02
496	-2.89869523E+02	4.06901659E-02
496	-2.60882571E+02	3.87823558E-02
496	-2.31895619E+02	3.65638279E-02
496	-2.02908666E+02	3.41376673E-02
496	-1.73921714E+02	3.16069594E-02
496	-1.44934762E+02	2.90747897E-02
496	-1.15947809E+02	2.66442434E-02
496	-8.69608570E+01	2.44184060E-02
496	-5.79739047E+01	2.25003627E-02
496	-2.89869523E+01	2.09931989E-02
496	0.	2.00000000E-02
567	-4.93812517E+02	2.00000000E-02
567	-4.69121892E+02	2.44524422E-02
567	-4.44431266E+02	2.80717702E-02
567	-4.19740640E+02	3.09187232E-02
567	-3.95050014E+02	3.30540407E-02
567	-3.70359388E+02	3.45384620E-02
567	-3.45668762E+02	3.54327264E-02
567	-3.20978136E+02	3.57975734E-02
567	-2.96287510E+02	3.56937422E-02
567	-2.71596885E+02	3.51819723E-02
567	-2.46906259E+02	3.43230029E-02
567	-2.22215633E+02	3.31775735E-02
567	-1.97525007E+02	3.18064234E-02
567	-1.72834381E+02	3.02702919E-02
567	-1.48143755E+02	2.86299185E-02
567	-1.23453129E+02	2.69460424E-02
567	-9.87625035E+01	2.52794030E-02
567	-7.40718776E+01	2.36907397E-02
567	-4.93812517E+01	2.22407919E-02
567	-2.46906259E+01	2.09902989E-02
567	1.17097443E-11	2.00000000E-02
637	-4.07885988E+02	2.00000000E-02
637	-3.87491689E+02	2.22289769E-02
637	-3.67097389E+02	2.41040995E-02
637	-3.46703090E+02	2.56452387E-02
637	-3.26308791E+02	2.68722652E-02
637	-3.05914491E+02	2.78050500E-02

637	-2.85520192E+02	2.84634639E-02
637	-2.65125892E+02	2.88673778E-02
637	-2.44731593E+02	2.90366625E-02
637	-2.24337294E+02	2.89911890E-02
637	-2.03942994E+02	2.87508280E-02
637	-1.83548695E+02	2.83354505E-02
637	-1.63154395E+02	2.77649272E-02
637	-1.42760096E+02	2.70591292E-02
637	-1.22365796E+02	2.62379272E-02
637	-1.01971497E+02	2.53211920E-02
637	-8.15771977E+01	2.43287947E-02
637	-6.11828982E+01	2.32806059E-02
637	-4.07885988E+01	2.21964966E-02
637	-2.03942994E+01	2.10963377E-02
637	1.53477231E-11	2.00000000E-02
708	-3.21959459E+02	2.00000000E-02
708	-3.05861486E+02	2.05269162E-02
708	-2.89763513E+02	2.10591665E-02
708	-2.73665540E+02	2.15866178E-02
708	-2.57567567E+02	2.20991369E-02
708	-2.41469594E+02	2.25865906E-02
708	-2.25371621E+02	2.30388458E-02
708	-2.09273648E+02	2.34457693E-02
708	-1.93175676E+02	2.37972279E-02
708	-1.77077703E+02	2.40830886E-02
708	-1.60979730E+02	2.42932180E-02
708	-1.44881757E+02	2.44174830E-02
708	-1.28783784E+02	2.44457506E-02
708	-1.12685811E+02	2.43678874E-02
708	-9.65878378E+01	2.41737604E-02
708	-8.04898648E+01	2.38532364E-02
708	-6.43918918E+01	2.33961822E-02
708	-4.82939189E+01	2.27924646E-02
708	-3.21959459E+01	2.20319505E-02
708	-1.60979730E+01	2.11045067E-02
708	4.09272616E-12	2.00000000E-02
779	-2.36032930E+02	2.00000000E-02
779	-2.24231284E+02	1.96745079E-02
779	-2.12429637E+02	1.95025927E-02
779	-2.00627991E+02	1.94643688E-02
779	-1.88826344E+02	1.95399503E-02
779	-1.77024698E+02	1.97094514E-02
779	-1.65223051E+02	1.99529864E-02
779	-1.53421405E+02	2.02506695E-02
779	-1.41619758E+02	2.05826149E-02
779	-1.29818112E+02	2.09289369E-02
779	-1.18016465E+02	2.12697497E-02
779	-1.06214819E+02	2.15851675E-02
779	-9.44131720E+01	2.18553045E-02
779	-8.26115255E+01	2.20602749E-02
779	-7.08098790E+01	2.21801931E-02

779	-5.90082325E+01	2.21951731E-02
779	-4.72065860E+01	2.20853293E-02
779	-3.54049395E+01	2.18307759E-02
779	-2.36032930E+01	2.14116270E-02
779	-1.18016465E+01	2.08079970E-02
779	2.50111043E-12	2.00000000E-02
850	-1.50106401E+02	2.00000000E-02
850	-1.42601081E+02	2.00000000E-02
850	-1.35095761E+02	2.00000000E-02
850	-1.27590441E+02	2.00000000E-02
850	-1.20085121E+02	2.00000000E-02
850	-1.12579801E+02	2.00000000E-02
850	-1.05074481E+02	2.00000000E-02
850	-9.75691606E+01	2.00000000E-02
850	-9.00638406E+01	2.00000000E-02
850	-8.25585205E+01	2.00000000E-02
850	-7.50532005E+01	2.00000000E-02
850	-6.75478804E+01	2.00000000E-02
850	-6.00425604E+01	2.00000000E-02
850	-5.25372403E+01	2.00000000E-02
850	-4.50319203E+01	2.00000000E-02
850	-3.75266002E+01	2.00000000E-02
850	-3.00212802E+01	2.00000000E-02
850	-2.25159601E+01	2.00000000E-02
850	-1.50106401E+01	2.00000000E-02
850	-7.50532005E+00	2.00000000E-02
850	2.72848411E-12	2.00000000E-02

APPENDIX D

RESULTS FOR TRAPEZOIDAL FLAT PLATE

In order to check the validity of the structural analysis, this appendix presents a comparison of theoretical and experimental natural frequencies of vibration of a cantilevered trapezoidal plate. The theoretical flutter speed and frequency for that plate are also presented but are not compared with experimental results. The dimensions and material properties of the aluminum plate are given in figure 12. The results are summarized in table I where theoretical vibration results are shown from SWIFT, which is the computer program based on the analysis presented in this report and described in appendix C, and from NASTRAN, a general-purpose finite-element structural-analysis computer program. The experimental results and the theoretical results obtained by using NASTRAN are taken from reference 26.

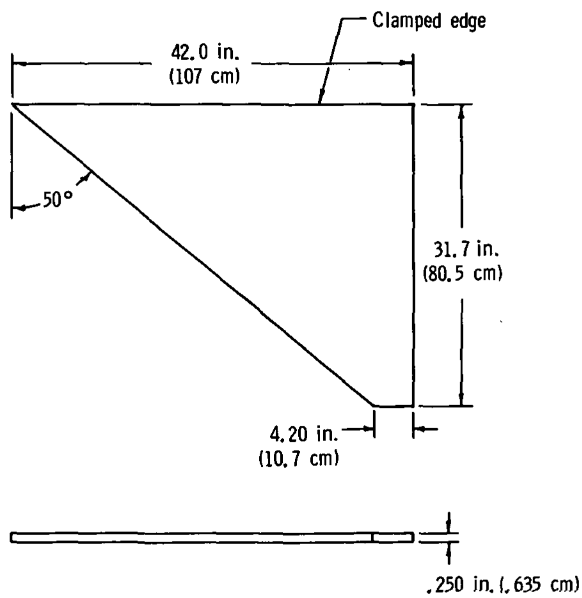


Figure 12.- Dimensions and material properties of aluminum plate used to obtain results given in table I.
$E = 1.00 \times 10^4$ ksi (68.9 GN/m^2); $\mu = 0.30$; Density, 0.100 lbm/in^3 (2770 kg/m^3).

Achieving a true "fixed" boundary condition experimentally is very difficult. The support must be massive, and the specimen-attachment bolts must be numerous, heavy, and well-placed. The experimental results presented in table I are believed to approach a true fixed-boundary condition.

TABLE I.- EXPERIMENTAL AND THEORETICAL NATURAL FREQUENCIES AND THEORETICAL FLUTTER SPEED AND FREQUENCY OF CANTILEVERED TRAPEZOIDAL PLATE SHOWN IN FIGURE 11

Experimental natural frequency, Hz	Theoretical natural frequency, Hz, for –	
	NASTRAN (109 grid points with 327 degrees of freedom)	SWIFT (10 spanwise stations with 33 degrees of freedom)
12.3	12.1	12.1
38.6	41.8	42.9
63.5	63.7	61.4
95.0	96.9	107.0
151.3	147.7	150.2

Theoretical flutter speed . . . . 1180 m/s (3890 ft/s)
Theoretical flutter frequency . . . . 35.7 Hz

A comparison between the experimental results and both sets of theoretical results indicates that the structural model used in SWIFT is adequate for simple wing structures.

In order to provide some insight into the flutter calculations, curves showing the locus of points for which the real part of the flutter determinant is zero and curves showing the locus of points for which the imaginary part of the flutter determinant is zero are shown in figure 13. For this case, the two flutter solutions shown in figure 13 are widely separated.

Following figure 13, a portion of the computer printout which contains the pertinent input data for the flutter calculations is presented. The input data used to obtain the natural frequencies differ only in that RHOA, the density of the air, was set equal to zero.

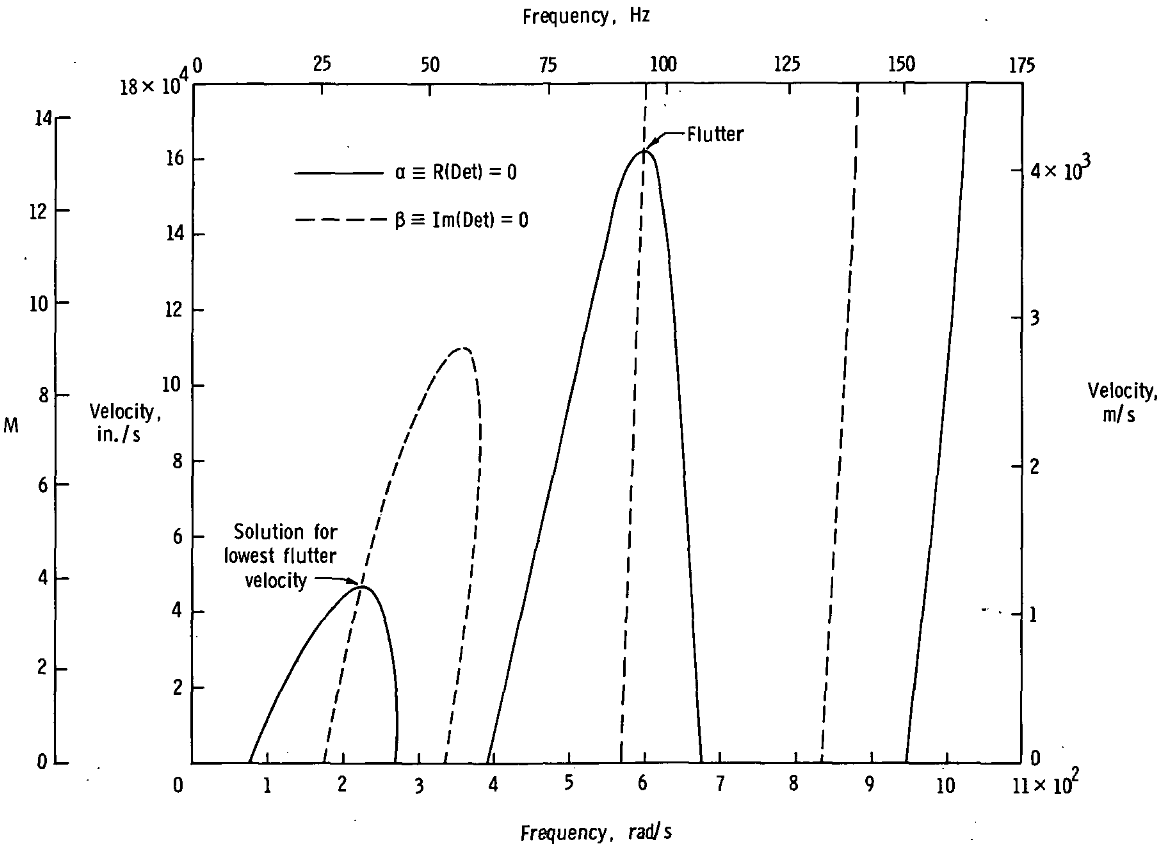


Figure 13.- Locus of points for which the real part of the flutter determinant, R(Det), is equal to zero (solid lines) and for which the imaginary part, Im(Det), is zero (dashed lines).

```
DETERMINATION OF THICKNESS DISTRIBUTION FOR MINIMUM WEIGHT WING
SATISFYING PRESCRIBED FLUTTER AND STRENGTH CRITERIA.
REFERENCE - NACA TN-3640,    STEIN-SANDERS,    JUNE,1956.
DEXTER-STROUD     RDK288     MAY,1969.

DATE 04/27/71

WING DESCRIPTION

ASPECT RATIO............AR =  2.74614000E+00
SURFACE AREA..........AREA =  7.32685000E+02 IN.**2
L.E. SWEEP ANGLE.......ANG =  5.00000000E+01 DEGREES
SEMISPAN................SS =  3.17179412E+01 IN.
ROOT CHORD..............RC =  4.20000028E+01 IN.
DEPTH RATIO.............DR =  C.
WING MATERIAL.............   ALUMINUM
MASS DENSITY..........RHOW =  2.58800000E-04 (LB.*SEC.**2)/(IN.**4)
YOUNGS MODULUS...........E =  1.00000000E+07 PSI
POISSONS RATIO..........MU =  3.00000000E-01

AIR PROPERTIES

SPEED OF SOUND...........A =  1.22040000E+04 IN./SEC. =  1.01700000E+03 FT./SEC.
MASS DENSITY..........RHOA =  5.14000000E-08 (LB.*SEC.**2)/(IN.**4)
CORRESPONDING ALTITUDE.... =  2.50000000E+04 FT.

                    XXX
                    XXX
                    XXX     FIXED BOUNDARY
                   .XXX     CONDITION
                  . XXX
                 .  XXX
                .   XXX
               .    XXX
              .     XXX
             .      XXX
            .       XXX ROOT CHORD =  42.0 IN.
           .        XXX
          .         XXX
         .          XXX
        .           XXX
       . 50 DEG.    XXX
       . . . . . . .XXX
                    XXX
SEMISPAN =          XXX LEADING EDGE = A+B*Y
 31.7 IN.           XXX        A = -4.20000028E+01
                               B =  1.19175359E+00

EQUATION FOR THICKNESS - TBAR    (INCHES)

TBAR =    C( 1) + C( 2)*A + C( 3)*A**2 + C( 4)*A**3
       +(C( 5) + C( 6)*A + C( 7)*A**2 + C( 8)*A**3)*B
       +(C( 9) + C(10)*A + C(11)*A**2 + C(12)*A**3)*(1-B*B)
       +(C(13) + C(14)*A + C(15)*A**2 + C(16)*A**3)*(1-B*B)*B

       WHERE  A = 1-Y/SS   AND   B = 1-2*X/(LE+TE)

                    (ITHICK = 2)

DESIGN VARIABLES            FINITE DIFFERENCE INCREMENTS     ANALYTICAL DERIVATIVES         * INDICATES ACTIVE
                                                                                              DESIGN VARIABLES
C( 1) =  2.50000000E-01     DC( 1) =  0.                     AD( 1) =  1.46385774E+02
```

```
STATIONS

    NUMBER OF SEMISPAN STATIONS..................NN = 10
    NUMBER OF CHORDWISE STATIONS = (10 X NCS)...NCS =  1

INPUT OPTIONS

       IOPT = 1      1 - ITERATE CN OMEGA AND V, 2 - COMPLEX SEARCH
     IPRINT = 2      0 - MATRIX CCNSTRUCTION, 1 - SHORT PRINTOUT, 2 - INTERMEDIATE PRINTOUT, 3 - EXTENDED PRINTOUT
    ISNDWCH = 0      0 - SCLID WING, 1 - SANDWICH CONSTRUCTION
        IHP = 0      0 - THICKNESS VARIATION EFFECTS NOT ACCOUNTED FOR IN PISTON THEORY, 1 - THICKNESS EFFECTS ARE USED


INITIAL WEIGHT =  3.65964435E+01  POUNDS


RANGE AND INCREMENT OF ITERATION

    CMEGA INITIAL.............CMINIT =  3.20000000E+02 RADIANS/SEC.
    CMEGA INCREMENT...........DELCM =  2.00000000E+01 RADIANS/SEC.
    CMEGA FINAL...............CMFIN =  1.05000300E+03 RADIANS/SEC.

    V INITIAL.................VINIT =  0.             IN./SEC.
    V INCREMENT................DELV =  1.00000000E+04 IN./SEC.
    V FINAL....................VFIN =  2.00000000E+05 IN./SEC.
```

REFERENCES

1. Turner, M. Jonathan: Optimization of Structures to Satisfy Flutter Requirements. AIAA J., vol. 7, no. 5, May 1969, pp. 945-951.
2. Schmit, Lucien A., Jr.; and Thornton, William A.: Synthesis of an Airfoil at Supersonic Mach Number. NASA CR-144, 1965.
3. Ashley, H.; and McIntosh, S. C., Jr.: Application of Aeroelastic Constraints in Structural Optimization. Applied Mechanics, M. Hetényi and W. G. Vincenti, eds., Springer-Verlag, 1969, pp. 100-113.
4. Ashley, Holt: Aeroelasticity. Appl. Mech. Rev., vol. 23, no. 2, Feb. 1970, pp. 119-129.
5. Rudisill, Carl S.; and Bhatia, Kumar G.: Optimization of Complex Structures to Satisfy Flutter Requirements. AIAA J., vol. 9, no. 8, Aug. 1971, pp. 1487-1491.
6. Sheu, C. Y.; and Prager, W.: Recent Developments in Optimal Structural Design. Appl. Mech. Rev., vol. 21, no. 10, Oct. 1968, pp. 985-992.
7. Morrow, William M., II; and Schmit, Lucien A., Jr.: Structural Synthesis of a Stiffened Cylinder. NASA CR-1217, 1968.
8. Thornton, William A.; and Schmit, Lucien A., Jr.: The Structural Synthesis of an Ablating Thermostructural Panel. NASA CR-1215, 1968.
9. Fox, R. L.; and Kapoor, M. P.: Structural Optimization in the Dynamics Response Regime: A Computational Approach. AIAA J., vol. 8, no. 10, Oct. 1970, pp. 1798-1804.
10. Giles, Gary L.: Procedure for Automating Aircraft Wing Structural Design. J. Struct. Div., Amer. Soc. Civil Eng., vol. 97, no. ST1, Jan. 1971, pp. 99-113.
11. Stroud, W. Jefferson: Minimum-Mass Isotropic Shells of Revolution Subjected to Uniform Pressure and Axial Load. NASA TN D-6121, 1971.
12. Kavlie, Dag; and Moe, Johannes: Automated Design of Frame Structures. J. Struct. Div., Amer. Soc. Civil Eng., vol. 97, no. ST1, Jan. 1971, pp. 33-62.
13. Comm. on Metric Pract.: ASTM Metric Practice Guide. NBS Handbook 102, U.S. Dep. Com., Mar. 10, 1967.
14. Stein, Manuel; and Sanders, J. Lyell, Jr.: A Method for Deflection Analysis of Thin Low Aspect Ratio Wings. NACA TN 3640, 1956.
15. Houbolt, John Cornelius: A Study of Several Aerothermoelastic Problems of Aircraft Structures in High-Speed Flight. Prom. Nr. 2760, Swiss Fed. Inst. Technol. (Zürich), 1958.

16. Morgan, Homer G.; Huckel, Vera; and Runyan, Harry L.: Procedure for Calculating Flutter at High Supersonic Speed Including Camber Deflections, and Comparison With Experimental Results. NACA TN 4335, 1958.

17. Fiacco, Anthony V.; and McCormick, Garth P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. John Wiley & Sons, Inc., c.1968.

18. Bracken, Jerome; and McCormick, Garth P.: Selected Applications of Nonlinear Programming. John Wiley & Sons, Inc., c.1968.

19. Moe, Johannes: Design of Ship Structures by Means of Non-Linear Programming Techniques. Symposium on Structural Optimization, AGARD CP No. 36, Oct. 1970, pp. 4-1 – 4-29.

20. Fox, Richard L.: Optimization Methods for Engineering Design. Addison-Wesley Pub. Co., Inc., c.1971.

21. Pierre, Donald A.: Optimization Theory With Applications. John Wiley & Sons, Inc., c.1969.

22. Wilde, Douglass J.; and Beightler, Charles S.: Foundations of Optimization. Prentice-Hall, Inc., c.1967.

23. Pope, G. G.; and Schmit, L. A., eds.: Structural Design Applications of Mathematical Programming Techniques. AGARDograph 149, Feb. 1971.

24. Anon.: Control Data 6000 Computer Systems Matrix Algebra Subroutines Reference Manual. Publ. No. 60135200, Control Data Corp., June 1966.

25. Hague, Donald S.; and Glatt, Curtis R.: An Introduction to Multivariable Search Techniques for Parameter Optimization (and Program AESOP). Contract No. NAS 2-4507, Boeing Co., Apr. 1968. (Available as NASA CR-73200.)

26. Clary, Robert R.: Practical Analysis of Plate Vibrations Using NASTRAN. NASTRAN: Users' Experiences, NASA TM X-2378, 1971, pp. 325-341.

NASA-Langley, 1971 — 32 L-7876

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

U.S.MAIL

POSTMASTER: If Undeliverable (Section 1: Postal Manual) Do Not Ret

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546